SUPER ROBONOVA RETURNS • VAPOR BOTS • POWER TOOL RACING

SERVO MAGAZINE

FOR THE ROBOT EXPERIMENTER

www.servomagazine.com

JULY 2006

AUTONOMOUS ROBOTS and Multiple Sensors

- Building H-Bridges
- Robotic Arms and Grippers
- Beyond Bleeps and Bloops

SERVO

ON THE COVER

44 **Autonomous Robots and Multiple Sensors**
by Bryan Bergeron
Part 1: Fusion Fundamentals

Features & Projects

36 **Mobility to the Maxx**
by Chris Cooper
Part 4: A Sense of Direction.

40 **Power Tool Drag Racing**
by Simone Davalos
Tim the Toolman meets Jeff Gordan.

51 **Building (H-)Bridges**
by Peter Best
Learn how to build electronic circuitry that controls myriad functions within a motor's magnetic domain.

ENTER WITH CAUTION:

28 **The Combat Zone**

SERVO Magazine **(ISSN 1546-0592/CDN Pub Agree#40702530)** is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL ENTRY MAILING OFFICES. POSTMASTER: Send address changes to **SERVO Magazine, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@servomagazine.com

07.2006
VOL. 4 NO. 7

Departments

06 Mind/Iron
07 Bio-Feedback
24 Events Calendar
25 Robotics Showcase
26 New Products
35 Robo-Links
74 *SERVO* Bookstore
76 Menagerie
82 Advertiser's Index

Columns

08 **Robytes** *by Jeff Eckert*
Stimulating Robot Tidbits

10 **GeerHead** *by David Geer*
The Crusher Military Robot Prototype!

14 **Ask Mr. Roboto** *by Pete Miles*
Your Problems Solved Here

18 **Twin Tweaks** *by Bryce and Evan Woolley*
Super Robonova Returns

57 **Rubberbands and Bailing Wire** *by Jack Buffington*
How to Record and Play Back Any Sound

60 **Programmable Logic** *by Gerard Fonte*
Xilinx vs. CPLD

64 **Robotic Trends** *by Dan Kara*
Money Talks, Coolness Walks

67 **TidBOTS** *by Dave Prochnow*
More Exciting Robot News!

68 **Robotics Resources** *by Gordon McComb*
Robotic Arms and Grippers

72 **Brain Matrix** *by Pete Miles*
Three Servo Hexapod Robot Kits

77 **Appetizer** *by Jonathan Fant*
The Robots Are Here! ... Well, Almost

79 **Then and Now** *by Tom Carroll*
Robots Who Care for People

Coming 08.2006

The FaceWalker

Mind / Iron

by Gary Mauler

Wow, it was a great feeling that night knowing that our Robot Fest was once again a success and that we were done for another year. But even as our group of dedicated volunteers celebrated another successful Robot Fest over dinner, the conversation quickly turned to what we could do better next year.

How quickly we forget the weeks of hard work and stress that go into planning, promoting, and running a Robot Fest! The one question that I am always asked is, Why do you keep volunteering each year? I usually ask myself that same question about four days before the event when I am totally stressed out!

This was the sixth year for our Robot Fest. As usual, the weeks leading up to the event were exciting when we signed up a new robot team or group to attend the event but there were also the times when we received bad news that someone had to cancel. Then there is always the worry that no one will attend or that there will not be enough robots on display for the public.

Volunteering to run an event can be an emotional roller coaster ride for all involved. The good news is that each time you go through the process, it really does get easier. It just does not seem like it at the time.

To answer the question why we keep volunteering each year, you need to go back and answer the question, "Why did you start down this path in the first place?" For me, it was because I wanted to see the kids in our school have a new opportunity for a type of learning experience that our public school system was incapable of delivering alone.

After doing some research on the Web and then attending an event sponsored by MIT called "Mind Fest — A Day of Playful Invention" at the legendary Media Lab in Cambridge, MA, I wanted to see what would happen if I brought a similar gathering of robot techies to my community. I was quickly able to sell the idea of a Robot Technology Club to other parents, as well as the school administrators at my kids' school. I proposed the format as an after-school activity, focused on applying the "engineering process" to building robots.

Having met the "fathers" of the LEGO Robotics Invention System (Dr. Fred Martin, Dr. Mitchel Resnick, and Dr. Semor Papiet) at the MIT Media Lab, I quickly decided to purchase these innovative construction kits for our Technology Club.

From that point on, I was "involved." It was rewarding to see how much enjoyment the kids got from building LEGO robots. I experienced a lot of personal satisfaction each time a fifth grader expressed how much fun they had building robots. The real test of how successful our Technology Club was would come at 4 pm each Wednesday when the workshop was supposed to end for the day. But, instead of our kids waiting for the bell to ring, we actually had to boot them out the door!

Another surprising measure of our Club's success was when teachers approached me with a look of total amazement and told me that they had "never seen kids so interested in what they were doing" and that for some reason they could not figure out why I never seemed to have a discipline problem at our Club meetings.

I guess we were doing something right! My feelings of satisfaction from so much positive feedback only made me want to volunteer more. The one question that I would always ask myself was, "How could I make a *geeky* thing like the Technology Club COOL for the kids?"

The idea that we came up with was to organize a Robot Fest for all my "geeky" kids at the end of each school year. I reasoned that the other kids who participated in sports and other activities were constantly recognized in the local media and at school. I wanted to provide a day for my kids to "be cool" — a day, where they could feel important and receive positive feedback from their peers and the general public that would attend the Robot Fest.

I was fortunate to be associated with an elementary school located in a high

Mind/Iron Continued →

Published Monthly By
T & L Publications, Inc.
430 Princeland Court
Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Product Order Line **1-800-783-4624**
www.servomagazine.com

Subscriptions
Inside US **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277
North Hollywood, CA 91615

BIO-FEEDBACK

Dear SERVO:

Great magazine guys. I am going to get into robotics and your magazine is fantastic! I have some suggestions regarding the reader who wondered how he could remove a broken tap.

All of the 10 suggestions Mr. Roboto gave were very good. Maybe I could add some more. I teach adults in a trade school in Anjou, Quebec (in Montreal), sometimes in the machinist course but mostly the CNC course.

Additional suggestions (to continue the previous list): 11) Buy and use only machine taps (not hand taps). These are sometimes named gun taps or spiral point taps. The advantage of these is that you do not have to keep backing out the tap — just keep on going. They don't cost much more than the hand taps and are much easier to use. (For example, at KBC Tools a 3/8-16 manual tap sells for $3.60 CAN and a spiral point tap sells for $4.88 CAN.)

12) Make yourself an alignment block. This can be any small piece of scrap steel (say, 3/4” by 1” by 1/2” thick) in which you drill a series of holes that are simply slide-fit holes for all the taps you will be using (say #4 up to 3/8”). As an example, you could drill a 1/4” hole for a 1/4” tap, etc. Ideally, you should drill these holes on a drill press (verify that the head of the drill press is reasonably square with the table). Then, when you wish to tap a hole (after you have drilled the proper hole — example a #7 drill for a 1/4-20 tap), just position your new alignment block over the hole to be tapped, hold it down with one hand, insert the tap in the appropriate hole, and tap away. The alignment block will keep the tap at right angles to the surface being tapped. This works even when tapping in awkward positions like vertical or overhead. Of course, the tap-drill has to be drilled square to the surface for this to work. I made one of these alignment blocks about 20 years ago and I still have it and use it in my basement workshop.

13) Buy yourself a ratchet-action T-handle. I bought two sizes — a small and a big one for about $20 each. After you have used one of these, you won't want to go back to the old T-handle!

14) To know the right size of drill for each tap, get a Tap-drill chart (usually free). I even typed the info that is contained on a tap-drill chart into my Zire Palm, so I always have the info at hand. I also compiled and entered into my Zire Palm charts for the sizes of various hardware (such as socket-head cap screws, etc.) and various handy formulas for calculating threads.

15) There was an article in the Oct/Nov 2002 issue of *Machinist’s Workshop* on how to make your own simple home-made EDM machine of the plunging type. When asking for a reprint at **www.homeshop machinist.net**, make sure you ask for the update information in the Dec

Continued on page 35

SERVO Magazine would like to sincerely apologize for two errors in Eric Scott’s article "Pneumatic System Safety" in the June issue on page 24. First, we inadvertently mis-spelled his last name as Stott. Secondly, the paren-ed comment " ... (just come to our house!)" was not written by Scott. It was an in-house comment that was never intended to get printed. Our bad ...

tech area of our state. The group of volunteer parents working with the Club would not have been successful without the support of the school staff. I was lucky to have talented, committed, and dependable volunteers to help each week at the Technology Club workshops. The ratio of students-to-parents was 5:1!

I truly believe this is what it takes to be successful when working with large groups of young students. The most amazing thing was that everyone was learning together how to build LEGO Mindstorm robots. Of course, the years of experience that my volunteers had in engineering and computer science ensured that the kids would be successful and not end up frustrated because they could not get something to work.

The really hard thing for the adults was to NOT build the robots for the kids. We made sure that they worked as a team and learned through their mistakes. The kids also learned that there were “many right answers” — a concept that was a little bit different from their normal school classes. The kids also learned that there is more than one right answer to solving problems.

The big thing for everyone to remember is the importance of avoiding “volunteer burnout.” This is the responsibility of every volunteer, as well as the parents of the kids who participate. For example, even if you as a parent can’t get off work to attend the club meetings or don’t have the technical skills to coach the kids, you still need to find some way to volunteer to help those who are devoting so much of their personal time to help your children.

Most people feel discouraged and put-upon if everything falls on their shoulders. After a year or two, they become frustrated and stop volunteering. Unfortunately, in my six years of running the Robot Fest, I have seen at least two groups where the two lead volunteers developed “volunteer burnout.” It was a disappointment because they had been doing such a great job and their kids were getting an experience of a lifetime. If they had received better support from other parents, I believe that they would have still been at it today. The message is “get involved.”

Volunteer leaders can also take steps to avoid their own “volunteer burnout.” There are two things that you can do is to evade this: delegate and train. I have seen too many volunteers try to do everything by themselves, mainly because “it is easier if I just do it myself.” It may seem that way, but there are probably some parents who would love to help if they only felt like they were welcome and needed. The other thing to remember is that as a volunteer, you need to constantly work on training your replacement so that you can move along with your own kids.

The last thought that I would like to leave with you, is that this robot technology that we read about each month in *SERVO Magazine* is truly a fantastic learning tool for your children and those in your community. The one thing that I like about it is that the kids who participate are actually learning a ton of real, lifelong skills that will give them a leg-up over their peers. But they think that they are just having fun. (We fooled them, didn’t we!?)

They learn how to work as a team, lead, experiment, innovate, solve problems from different perspectives, communicate, and persist in finding the best solution. These skills prepare them to become great engineers and inventors who will fulfill the never-ending need for technology in our society. **SV**

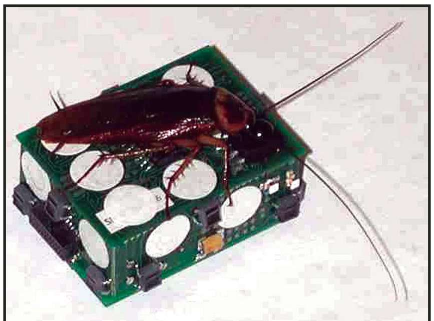
Robytes

by Jeff Eckert

Are you an avid Internet surfer who came across something cool that we all need to see? Are you on an interesting R&D group and want to share what you're developing? Then send me an email! To submit related press releases and news items, please visit www.jkeckert.com

— Jeff Eckert

Bots and Roaches Adapt to Each Other



A cockroach is strangely attracted to a tiny robot that has been coated with roach pheromones.

On a somewhat less appetizing level is some research conducted at the Université Libre de Bruxelles, in Belgium (**www.ulb.ac.be**). Developed under the European Commission's Future and Emerging Technologies (FET) Initiative and dubbed project Leurre, small insect-like robots ("insbots") were fitted with two motors, wheels, a rechargeable battery, several computer processors, a light-sensing camera, and an array of infrared proximity sensors. In an experiment, they were placed in a maze of curved walls wherein they proved their ability to navigate by avoiding the walls, obstacles, or each other, follow the walls, congregate around a lamp beam, and even line up.

When placed in the same area with cockroaches, the robots adapted their behavior by mimicking the animals' movements. And when coated with pheromones taken from roaches, the robots even fooled the insects into thinking they were real creatures, after which the roaches apparently began to imitate the behavior of the insbots. (Two side projects in the Leurre program also experimented with sheep and chickens, but we won't go there.)

And what's the point of all this? According to project coordinator Jean-Louis Deneubourg, "We believe farming in Europe can only survive if it is associated with high technology ... A robot interacting with animals, even if it is not mobile, could be used for numerous tasks, such as herding or milking. Our project demonstrates that the fields of biology and IT can work together more closely in the future." Details are available at **leurre.ulb.ac.be/index2.html**

Robotic Equipment Supports Minimally Invasive Surgery

William Peine — an assistant professor at Purdue University — operates hand controls for a surgical robot under development. Photo courtesy of Purdue News Service. Photo by David Umberger.

Robotic surgical devices (e.g., Intuitive Surgical's da Vinci system) are highly useful for minimally invasive surgery, but they are expensive and complicated. However, a mechanical engineer at Purdue University (**www.purdue.edu**) is working with doctors to come up with a system that will be less expensive (about $250,000), portable, and still versatile enough for a wide variety of operations.

The basic idea is an extension of laparoscopic surgery, in which a surgeon uses instruments inserted through small openings, thus, eliminating large incisions that leave scars and require a long recovery time.

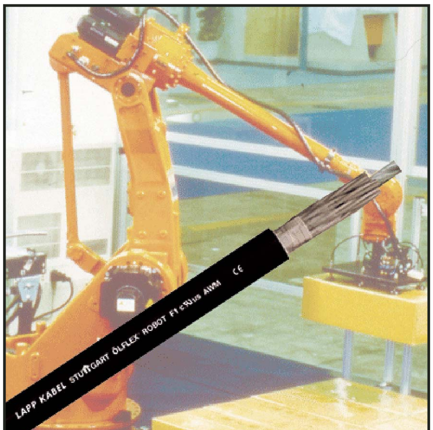
Without robots, surgeons manipulate the laparoscopic probes with handles that remain outside the body. Using hand-held tools can be tricky because it is difficult to manipulate the devices. For example, there is the "fulcrum effect" in which moving the handle in one direction causes a probe to move in the opposite direction inside the body. But a robotic device can compensate for the effect.

During robotic surgeries, the surgeon sits at a console and uses hand controls to direct robotic arms that move the probes and a camera lets the surgeon see inside the body during the operation. The camera magnifies the view on a computer screen mounted on the console. The researchers are also trying to incorporate tactile sensors into the robots to enable surgeons to "feel" tissue so as to better diagnose medical conditions and tie them to CT scanners, ultrasound equipment, and MRI devices for guidance.

The goal is to come up with a device that is suitable for such procedures as the treatment of prostate cancer, stomach surgery, and even operations on heart valves without the need for open-heart surgery. Apparently, the system will be marketed by a company called

Pressure Profile Systems (**www.presureprofile.com**), which already sells tactile sensitive devices.

Cable Designed for Continuous Twisting



Lapp USA has introduced a new cable for multi-axis robotic applications. Photo courtesy of Lapp.

If your latest project involves cables that must move and flex a great deal, you might want to take a look at the OLFLEX® ROBOT F1 UL/CSA from Lapp USA. Introduced at this year's National Manufacturing Week Show, it is designed to provide reliable mechanical performance on multi-axis robots, welding robots, and manipulators; to connect rotating and tilting tables; and in other applications requiring bending and torsion movements.

It is manufactured using flexible bare copper conductors, special polymer insulation, nonfriction tape, and an overall tinned copper braid shield, if needed. It also features an oil-, abrasion-, and spark-resistant polyurethane elastomer jacket and remains flexible through a temperature range of -40 to +80°C. To locate your nearest dealer, just visit **www.lappusa.com**

Stepper Drivers Available for Hobbyists

Recently introduced by LNS Technologies (**techkits.com**) is the MSD-62M stepper motor driver, designed for robots, CNC routers, engraving machines, security cameras, and a range of other build-it-yourself applications. It is based on the Allegro/Sanken SLA7062M IC chip, which combines low-power CMOS logic with high-current, high-voltage power FET outputs.

The MSD-62M provides versatility for a range of applications. Photo courtesy of LNS Technologies.

It is capable of handling motor winding currents of up to 3 A per phase, and it operates from a single supply voltage of 10 to 40 VDC. The drive works with any unipolar (six- and eight-wire) motor and is adjustable from 500 mA to 3 A via an onboard pot. LNS also offers the BSD-298, which works with bipolar (four- and nine-wire) motors.

Either one will run you $89.95 assembled and tested. Kit versions are also available. Neither comes with a power supply, which will run you another $129.95. **SV**

GEER HEAD

by David Geer
Contact the author at geercom@alltel.net

The Crusher Military Robot Prototype!

“Roads? It Don’t Need No Stinking Roads!”

Human Creature To Crusher Comparison

If you were to list the various capabilities we humans possess that make us capable of mobility in the most unique of environments, it might go something like this: we can think independently; we can sense our environment; we can plan our course of movement accordingly; and we can respond to obstacles and varying terrain by changing course and adjusting our weight, balance, and footing.

Military robots become more useful as they become capable of more of the things that we can do and, perhaps, even more than we can do. Crusher is the name of a recently created unmanned robot vehicle that fits the bill. Thanks to Professor John Bares of the Carnegie Mellon Robotics Institute (creators of the robot vehicle) and Carnegie’s Director of Business Development Steve DiAntonio, I can tell you all about it.

Crusher

Crusher is a one-of-a-kind, 6.5 ton metal alloy unmanned robot vehicle. Traveling on six wheels, it looks at first like an over-sized RC’d military toy. But this aluminum, titanium, and steel-bellied monster packs a wallop — at up to 26 mph — on anything that it lands on (watch the videos at the links listed in the sidebar) or that gets in its path (it perceives and avoids obstacles that are too big for it to tangle with).

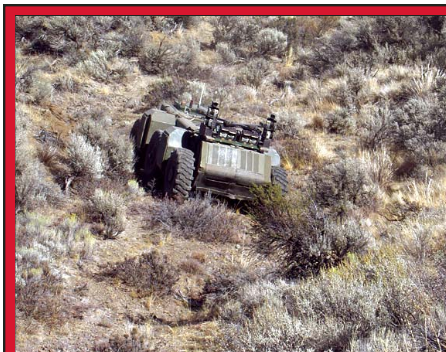
Crusher’s precision mobility is enabled by six separate embedded electric motors in each of its six wheels. The motors are powered by a hybrid power system of rechargeable batteries, and the turbo diesel generator that recharges them.

The hull (courtesy of CTC Technologies, Pennsylvania) is “high-test” (my slang for heavy-duty) aluminum tubing with titanium “nodes” with an outer skin (skid plate) of steel. This combination gives Crusher a high level of shock absorption from heavy impacts (you’ve really got to watch the videos).

The Irish engineered suspension (Timony Technology, Meath) gives Crusher a smooth ride despite the usual (or man-made for testing) off-road hazards of huge boulder piles, barriers, and gulleys.

If the Army ever needs Crusher to get really mean (as opposed to simply producing collateral damage), it can be fitted with more than 8,000 lbs of armor and weaponry payloads, as well as many other practical add-ons.

Crusher easily traveling down-hill through brush and vegetation.



Crusher crossing a creek — smooth sailing all the way!

Crusher can quickly increase its speed, even in very difficult terrain.

History — Beware the Rise of the Acronyms!

The roots of Crusher lie in its predecessor — Spinner — and in the Unmanned Ground Combat Vehicle (UGCV) and Perception for Off-road Robotics (PerceptOR) projects. The two DARPA-funded Army programs were combined into the UGCV PerceptOR Integration (UPI) project in May of 2004.

The first parent program — UGCV — started in 2001 offering up the R&D offspring known as Spinner — a durable, unmanned, off-road vehicle built with high performance for difficult, off-road terrain in mind. Like some motored toys you may have seen, Spinner was most unique in that it had continued mobility in an inverted position. You could turn it completely over and its wheels could still touch the ground and keep on trucking!

The second parent program — PerceptOR — which began that same year, prioritized sensing and autonomy over mobility in order to develop navigation skills for difficult off-roading challenges like trees, ditches, and boulders.

The combined UPI project, which produced Crusher, has been funded by DARPA and the US Army to the tune of over $35 million so far.

CRUSHER vs. SPINNER COMPARISON

Spinner — a prototype previous to Crusher — was named for its ability to run while inverted. Crusher is named for its ability to climb over boulders and other large obstacles.

Spinner — the six-wheeled older sibling to Crusher — is the unmanned ground combat vehicle (UGCV) that resulted from a Defense Advanced Research Projects Agency (DARPA) request to pour $5.5 million in funding into a prototype for all terrains.

The Carnegie Mellon National Robotics Engineering Center's (NREC) Spinner was unique in its ability to keep moving even if it was flipped upside down.

Other goals for the project included that it be easily teleoperated and able to hold up in moderate crash and recovery scenarios.

Spinner was proof enough of the potential of this course of research to motivate DARPA to fund the research and construction of its younger sibling — Crusher — based in part on Spinner.

The biggest difference between the two robots is that Crusher has a different durability-to-weight ratio. It is a much tougher vehicle, according to John Bares, Associate Research Professor in Carnegie Mellon's Robotics Institute and Director of the National Robotics Engineering Consortium.

Crusher is more durable than Spinner, despite weighing 30-percent less than its predecessor. It has a tougher "under belly" and better suspension than Spinner. It has a higher torque-to-weight ratio on the drive and a lot of modular improvements, as well.

Crusher payloads include additional and advanced sensors, fuel, and supplies, ambulatory payloads for carrying injured soldiers, and even weaponry and armor.

While Spinner was designed for inverted operation in case of flip-flops, Crusher was designed to simply avoid being turned over. Spinner's low center of gravity made it difficult for it to be turned over, anyway. By keeping Spinner's width and low center of gravity while ditching its ability to run inverted, researchers were able to dump some of the weight, cost, and complexity of the Spinner model for Crusher.

Crash Course in Crusher Mission Methodology

Still in research, Crusher is being tested with various payloads to determine the types of missions for which it will be optimal. Potential mission applications include: cargo vehicle, recon robot, soldier rescue, unmanned attack vehicle (with gun mount), and many more. It may be deployed in convoys that work in tandem to accomplish military objectives.

Feeling the Crush (Sensing)

Crusher — a near seven-ton unmanned vehicle — "drives by wire" using GPS waypoints to determine its next course of movement. It will soon be equipped with autonomous movement via various sensor packages. While there will always be communications between Crusher and a human operator, these will be limited to telling Crusher where to go — how it gets there is up to Crusher and its systems.

Developed for the military by the Carnegie Mellon University's Robotics Institute's National Robotics

Crusher surmounts a ditch, no problem.

Crusher drives through dusty atmosphere, unhindered.

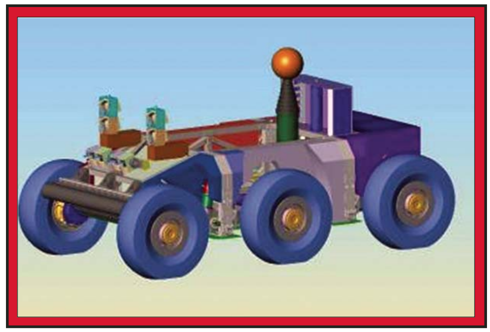
Crusher being tested at Fort Knox.

CARNEGIE MELLON ROBOTICS NEEDS YOU!

"We need great robotics engineers," says John Bares, Associate Professor, Carnegie Mellon Robotics Institute, " ... and I say that seriously in the sense that, for young people reading this, they need to go out and get a good math, science, and physics education to do well. For other people, give us a call — we're looking for great people."

Check out the Carnegie Mellon Robotics Institute and contacts at **www.ri.cmu.edu**

Field overview picture of Crusher.



3D model of Crusher's predecessor, Spinner.

Engineering Center, Crusher will, for example, be able to drive several meters and sense ditches, hills, humps, bushes, and trees on its own and determine whether it can go through or around them.

It will sense the terrain via laser sensor signals that go out, and by taking pictures of the terrain with digital cameras. The laser range finders send out about 75,000 pulses per second to measure distance.

The cameras are digital cameras that take a video image, digitize it frame-by-frame, and analyze the objects in the frame via the pixels to determine what size and type of object it may be. Based on the analysis of those pictures, it provides commands to its control system to guide its motors for movement.

Additional sensors on crusher include speed sensors on the motors, sensors on the suspension system that measure angles, pressure sensors on the suspension to measure force, and inertial sensors so it can "feel" and respond to the shock. There are about 1,000 sensors on the vehicle. They report the state of every component and the engine has all the sensors a normal engine would.

The sensors and computers mostly communicate via TCP/IP and UDP through a P2P (Point-to-Point) protocol.

Computer Brains and Programming

Crusher has its own computer brain that runs the navigation paths. It actually has several large and small computers for processing navigation decisions and sensor information.

Most of the programming of Crusher is written in C++. The main operating system is QNX — a real-time system for robot control.

Beyond the basic software used for the robot's actuators, steering, and brakes, there is software that processes the sensor images so that it can analyze them for size and distance to determine whether to go around or through them.

There are two approaches to image processing here. Traditionally, you would program in all the intelligence about the terrain that is available and the robot would be limited to working from that to determine an appropriate course of action.

For example, you would program in data that would determine whether Crusher is looking at a tree, boulder, ditch, and so on. You would program in information that the robot would use as its basis for determining whether the obstacle was sufficient to warrant a change of course.

This method of programming and processing sensor image data requires that you model the outside world — any potential environments the robot might face — and that requires a lot of code.

An optional approach that may be taken at some point is to program in the capacity to learn. If the robot can learn from its mistakes — learn which obstacles it should avoid next time — it can, to this degree, do its own programming of a sort and you avoid coding in every potential obstacle at the start.

Future Plans

There are two courses that the UPI, Spinner, and Crusher work could take. It could continue on in research, which could make room for another "design cycle" and further upgrades and improvements, according to DiAntonio.

From there it would proceed to

SO, I ASSUME IT CRUSHES THINGS?

Crusher is an unmanned vehicle and, yes, it crushes things. However, this is only as a by-product of its intended purpose. Crusher was built to survive and keep moving against all terrain related odds. With these capabilities, and the option to add a wide variety of payloads, it can be adapted to varying field work that is usually performed by operational personnel. This, in turn, keeps soldiers from being put at risk for those tasks.

The 6.5 ton, six-wheeled juggernaut prototype is stronger, more mobile, and soon to be more autonomous than other experimental prototypes of its size and nature.

third party research in autonomy and mobility, as well as other research by other organizations. There is a lot of work that could be done to even further advance Crusher's off-road readiness.

Or, the Army could at any point decide that Crusher is ready to go into production for military applications.

According to Bares, there is also a third potential path for Crusher and its kind. The Carnegie Mellon Robotics Institute involves itself in both military and commercial robotics research. It may well be that we'll see Crusher in some commercial application before we see it in combat.

"We're trying to get these systems into commercial use in agricultural fields and mining," says Bares. This would be a great opportunity for Carnegie Mellon to get feedback on how such vehicles operate in the field as production models.

It is also possible that the individual technologies growing in research and inside Crusher may be inherited by other projects.

Whichever course Crusher takes, in the mean time, other systems will likely be built in parallel with Crusher that will become more and more autonomous.

See Crusher in Action

Crusher is a very large, aggressive (as you'll judge from the videos), yet quiet unmanned vehicle. It is very smooth as it moves across tough terrain. Crusher has been tested at the National Robotics Engineering Center in Pittsburgh, PA, most frequently at a site off the beaten path in Somerset County.

If you are a government employee, a contractor associated with Future Combat Systems (FCS), or other robotic programs, you are invited to contact UPI program manager Dr. Larry Jackel at ljackel@darpa.mil to make arrangements to observe UPI field trials. SV

RESOURCES

Crusher site
www.rec.ri.cmu.edu/projects/crusher

Crusher videos
www.rec.ri.cmu.edu/projects/crusher/videos/index.htm

Numerous other CMU robotics projects
www.rec.ri.cmu.edu/projects

The QNX operating system
www.qnx.com

CMU Robotics Institute and contacts
www.ri.cmu.edu

Our resident expert on all things robotic is merely an Email away.

roboto@servomagazine.com

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

ASK MR. ROBOTO

by **Pete Miles**

Q. What does vapor bot mean? I have seen it mentioned in this magazine a couple times and I haven't been able to figure out what it means.

— Tim Caufman

A. Ah, the nebulous Vapor Bot. I also hear about them all the time and have yet had the privilege of actually seeing one. Well, the Vapor Bot is kind of like its name — a Vapor. A collection of gasses that floats around, but is not really there. A Vapor Bot is actually a robot that hasn't quite been completed. Well, in most cases, its construction hasn't even started, and many times these robots are just ideas and a box of parts.

What makes Vapor Bots different from other uncompleted robotic projects is that the creators of the Vapor Bot talk about their robots to other robot builders as if it is a built (or almost a completed) robot. They will compare specifications, capabilities, materials, power sources, technical and fabrication issues, and performance. They will usually brag about how well their robot is going to perform in the next competition and how it will be able to beat certain other robots. When the contest comes, the robot builder shows up, and the mysterious Vapor Bot doesn't materialize.

Now, don't misunderstand me in thinking that I am being critical of Vapor Bots and those who build them. I'm not. I have about a dozen Vapor Bots for every robot that I actually get around to building. This is probably true for most — if not all — robot builders. All robots begin as a Vapor Bot. This is the beginning of the idea. There are many reasons why a robot doesn't get built or completed, but that doesn't mean that the robot isn't real in the builder's mind's eye.

Q. My 10-year-old son loves everything about robots and wants to learn how to build his own. Can you recommend anything that would be good for a 10-year-old?

— Beth Porter

A. Well, you have several choices here. If your son is interested in building things that look like robots, walk around like robots, and are simple to build, take a look at the simple robotics kits that are made by Tamiya. These are simple plastic kits that are easy to assemble. There are two-, four-, and six-legged walkers and some two-wheeled robots, and most can be purchased for less than $20.

These kits are made out of clear plastic so you can see how all of the internal mechanisms work. They use a single AA battery to run the robot's electric motor that drives a gearbox

Kit Name	Locomotion	Part Number
Mechanical Kangaroo	Two-Legged	71102
Mechanical Ostrich	Two-Legged	71104
Boxing Fighter	Two-Legged — Remote Control	71107
Mechanical Rabbit	Four-Legged	71108
Mechanical Tiger	Four-Legged	71109
Mechanical Pig	Four-Legged	71111
Mechanical Beetle	Four-Legged	71103
Mechanical Racing Horse	Four-Legged	71112
Mechanical Dog	Four-Legged	71101
Mechanical Giraffe	Four-Legged	71105
Mechanical Turtle	Four-Legged	71106
Mechanical Insect	Six-Legged — Remote Control	71107
Wall Hugging Mouse	Two-Wheeled — Advanced	70068
Line Tracking Snail	Two-Wheeled — Advanced	75020
Mechanical Blow Fish	Two-Finned — Swimming	71114

Table 1. Tamiya's Robot Kits.

that moves a set of linkages that causes the legs to move. There is a lot of learning potential from these kits in that you can study how they work and your son can make copies of them to make larger-sized robots.

My first robot was a one-foot tall copy of a simple windup tin robot. Table 1 shows a list of the different kits that Tamiya manufactures. These kits are available at most of your local hobby and toy stores for $15 to $30 each. For more information about these kits, visit Tamiya's website at **www.tamiyausa.com** and use a key word search of Robot Kits. The two Remote Control kits would enable more interactions with your son and offer the potential for a lot of modifications to add more capabilities (see Figure 1).

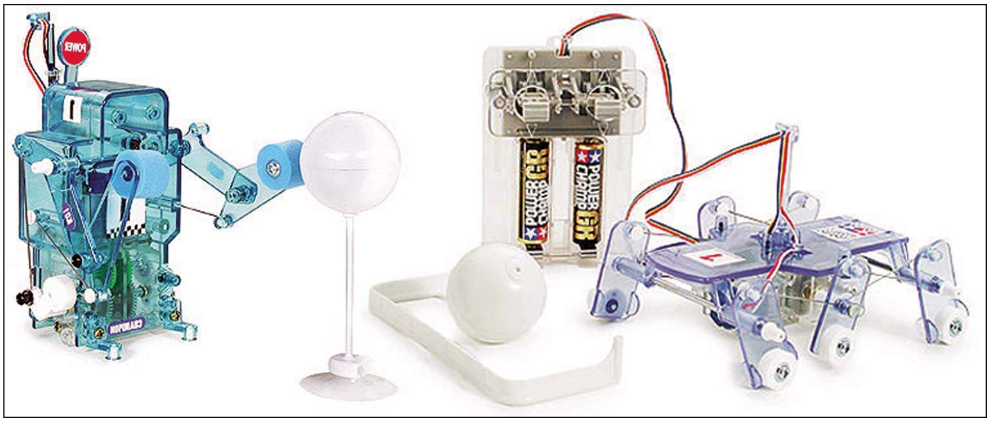


Figure 1. Tamiya's remote control Boxing Fighter and Mechanical Insect.

Now, for a more advanced robot kit — which I highly recommend — there is the LEGO Mindstorms Invention system (**www.legomindstorms.com**). If your son is already playing with LEGOs, then the Mindstorms invention system is the next natural progression for him. With this system, you can build just about any type of robot your imagination can come up with. The LEGO Mindstorms system uses regular parts and has some special LEGO sensors, motors, and a microcontroller (brain) called the RCX brick. The RCX brick can control three different outputs (motors) and can read in three different sensor inputs.

The RCX brick is programmed with a PC using a graphical-based programming language which is very intuitive and easy to learn. Each programming sequence is like a LEGO brick, and the program is snapped together like a regular LEGO structure. The CD that comes with the kit has a step-by-step guided tour that teaches how to build three different robots and how to program them. Within a few hours, your son will have a robot built from scratch — programmed and following a black line on the ground.

The LEGO Mindstorms Invention System is so popular and effective as a learning tool, an international robotics competition called FIRST LEGO League (**www.firstlegoleague.org**) was created to use them to help teach 9-14 year olds about science and technology. There are several thousand school teams that compete against each other every year in the September to December time-frame. If your son's school doesn't have a team, then talk to one of the school's administrators to get one started. I have been judging these events for a few years now, and it is absolutely amazing what the kids come up with using LEGOs and how they uniquely solve each of the challenges.

A point to note here — there is going to be a new version of the LEGO Mindstorms kit that will be released in August 2006. It is called the LEGO Mindstorms NXT. It has more capabilities than the original Mindstorms Invention System, and different types of sensors and motors. Two of the big changes are in the motors and programming environment. The motors in the NXT system can be either continuous rotating — like with the Invention System — but they have the ability to move to a specific position and hold there, much like a model airplane servo. It still has a graphical-based programming environment, but it is more like a wiring diagram (based on LabView, **www.ni.com**) which is also very intuitive to learn and use.

After August, you should be able to find both sets at major department stores that have the larger LEGO selections — such as Toys R Us — or you can purchase them from the Internet. I haven't seen pricing for the NXT yet, but I have heard that it is going to be in the same price range of the regular Invention System. Table 2 shows a simple comparison between the two LEGO sets. For anyone getting started in the world of robotics, there is no better way than to get started using the LEGO robotic systems described here.

Q. Is it possible to connect a Playstation 2 controller to my robot so I can drive it around?

— Seth Carson
Minneapolis, MN

	LEGO Mindstorms Invention System 2.0	LEGO Mindstorms NXT
Microcontroller	RCX Brick	NXT Brick
Motors	Two continuous DC motors	Three servo motors (continuous and position control)
Sensors	Two Touch, One Light	One Ultrasonic, One Sound, One Touch, One Light
Number of Inputs	Three	Four
Number of Outputs	Three	Three
Programming Interface	Infrared	USB
Number of LEGO Pieces	718	516

Table 2. Comparison between the LEGO Mindstorms Invention and NXT systems.

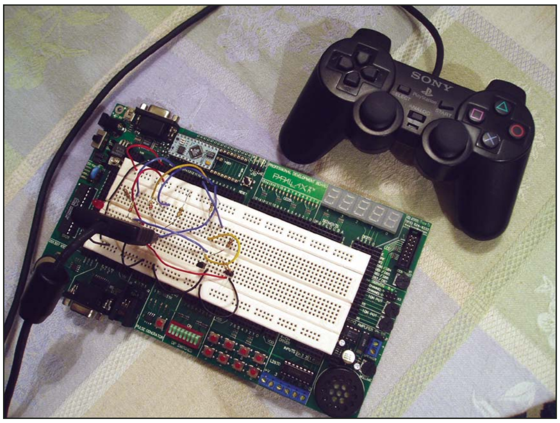


Figure 2. Playstation Dual Shock 2 controller wired to a BASIC Stamp.

A. Actually, this is not hard to do, and I am surprised that more people aren't already doing this. The Playstation 2 Controller makes an excellent robot controller since it has 14 digital switches and four analog axes. With this, you can control almost any feature on a robot. The only hard part you are going to have using this controller is finding the right connector for your controller to plug into.

For some background information, take a look at two articles published in *Nuts & Volts Magazine* by Aaron Dahlen — June '03 and Jon Williams — September '03. Aaron's article showed how to use a Playstation controller to control a Lynxmotion (**www.lynxmotion.com**) five-axis robotic arm and Jon's article introduced some improvements in the overall timing issues of the controller along with providing a more generic code for using the Playstation controller. Their articles showed me how to get a BASIC Stamp and the Playstation controller to talk to each other (see Figure 2).

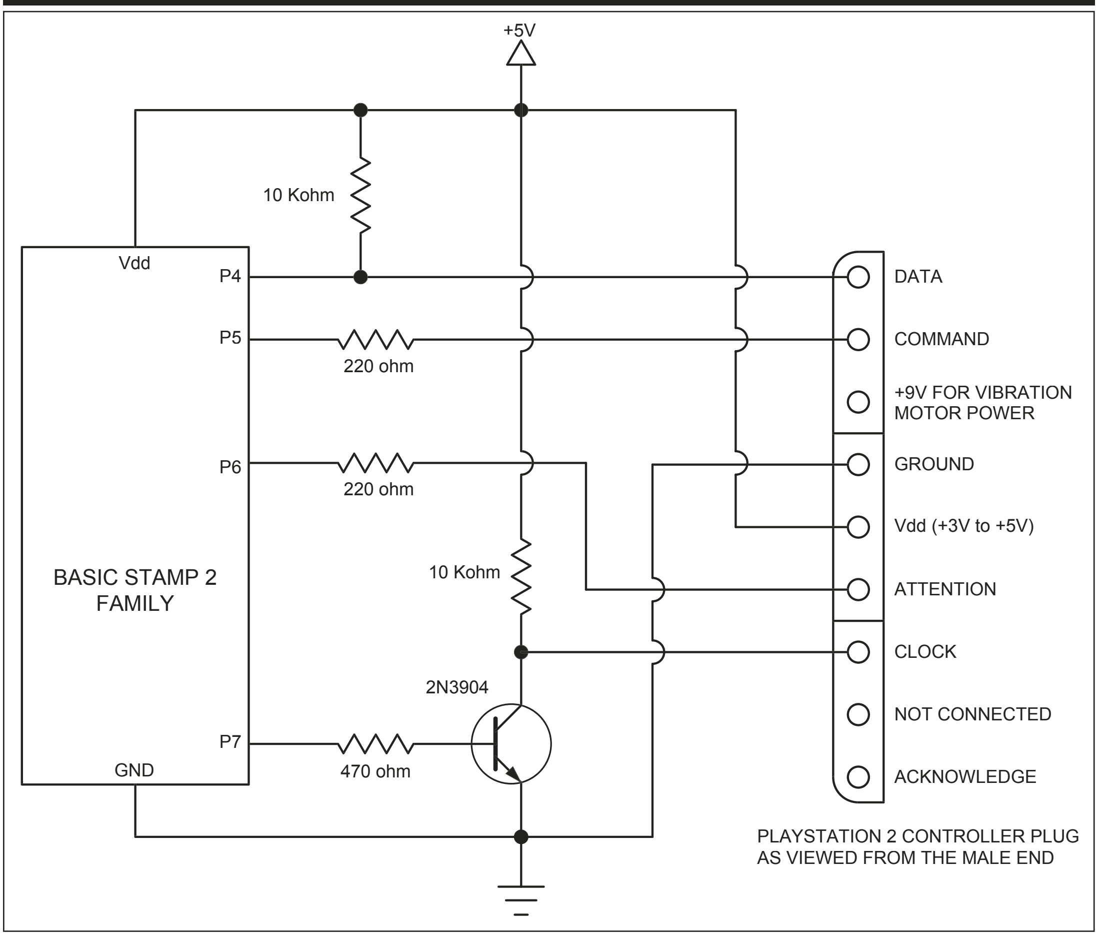
The first thing you need to do is build a simple electrical interface for the controller. Aaron introduced the concept of using a transistor to invert the clock signal from a BASIC Stamp to the Playstation controller so that the BASIC Stamp's SHIFTIN and SHIFTOUT commands can be used to simplify the programming. Figure 3 shows my version of this circuit. This interface circuit can be built by removing the transistor and the 10K resistor and directly connecting the clock signal line from the controller directly to the 470 ohm resistor. But if this is done, then a manual method will be needed to toggle the clock line while reading in each bit of data from the data line.

It turns out that not using hardware to invert the clock signal has a significant effect on the amount of time it takes to read in the data from a Playstation controller. Using the SHIFTIN and SHIFTOUT commands works well for reading in all the data from the buttons and three of the four joystick positions. But bit 7 of data from the y-axis of the left joystick is always set high due to the way the SHIFTIN command works and how the controller releases the data line. Jon's example code solves this problem by manually generating the clock signal for the last byte of data. So a combination of SHIFTIN, SHIFTOUT, and manually toggling the clock line while reading in the data signals ensures reading in all the data from the controller accurately.

The manual method of toggling the clock line while reading in each bit of data from the controller works well, but there is a time penalty. With the BASIC Stamp, it takes about 3.5 times longer to work with all 10 bytes of data that are transmitted between the Stamp and the Playstation controller when using the manual method over a combination of using the Stamp's built-in SHIFTIN, SHIFTOUT commands.

With a regular BASIC Stamp 2, it takes about 145 ms to read the controller using a pure manual method,

Figure 3. Electrical schematic for wiring a Playstation controller to a BASIC Stamp.



			DATA							
Byte	**Cmd**	**Variable**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
01	0x01	PsxOut	N/A							
02	0x42	PsxOut/PsxIn	0x41 in Digital Mode, 0x73 in Analog Mode							
03	N/A	PsxStatus	0x5A							
04	N/A	PsxThumbL	Left Arrow	Down	Right	Up	Start	N/A	N/A	Select
05	N/A	PsxThumbR	Square	X	O	Triangle	R1	L1	R2	L2
06	N/A	PsxJoyRX	Right Joystick: Left = 0, Neutral = 127, Right = 255 X-Axis							
07	N/A	PsxJoyRY	Right Joystick: Up = 0, Neutral = 127, Down = 255 Y-Axis							
08	N/A	PsxJoyLX	Left Joystick: Left = 0, Neutral = 127, Right - 255 X-Axis							
09	N/A	PsxJoyLY	Left Joystick: Up = 0, Neutral = 127, Down = 255 Y-Axis							

Table 3. Button and joystick position mapping from the Playstation controller.

and it takes about 40 ms to read in the data using the SHIFTIN and SHIFTOUT command approach. This is a good example of why using some additional hardware along with some built-in commands from a microcontroller can greatly improve the overall cycle timing of a project.

Depending on your robot, 40 ms may be too long. For example, many robots use model airplane servos as drive motors and joint actuators, and these servos require their position to be updated every 20 ms (unless a dedicated servo controller is being used.) When there is a need for speed, I like to use the BASIC Stamp 2px24. With this Stamp, the amount of time needed to read the Playstation controller is only 9 ms, which is over four times faster than a regular BASIC Stamp 2.

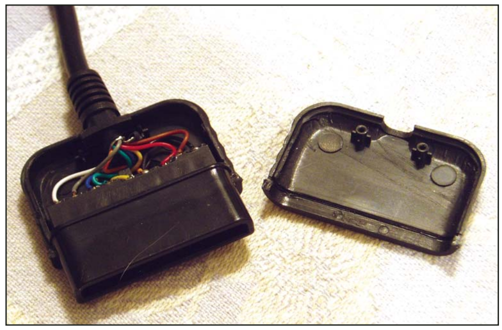
The sample program (available on the *Nuts & Volts* website, **www.nutsvolts.com**) is a simplified version of the program that Jon Williams published in his article in September '03. The subroutine named PSX_TxRx is the manual method for toggling the clock line while reading in each bit of data for a single byte. Table 3 maps which button position with the DATA results from the controller, along with the variable name from the program example. All of the buttons are active low.

As mentioned earlier, the hardest part of using a Playstation controller is probably finding a plug for the controller since this is a non-standard and proprietary plug design. I found a six-foot extension cable made by Intec (**www.inteclink.com**) at a local department store for $5. The cover is easily popped off by pushing a small screwdriver in the seam between the top and bottom covers.

Figure 4 shows the internal wiring inside this connector. The existing wiring can be removed and replaced with your own cable. Or, the existing cable can be cut somewhere along its six-foot length, and the wires from the cut end of the cable connected to your own connector.

The pin spacing in this connector is 0.156 inches, and the nine pins are divided into groups of three pins separated by a divider wall. A set of three-pin female connectors with 0.156 inch spacing can also be used to plug into the connector. **SV**

Figure 4. Internal view of a Playstation connector.



RESOURCES

Playstation 2 wiring
http://pinouts.ru/data/playstation_9_pinout.shtml

Aaron Dahlen's article and Jon Williams' article
www.nutsvolts.com

Last time, we began to equip the Robonova-1 from Hitec with an exosuit so it could complete scaled-down versions of the Tetsujin challenges. The three Tetsujin challenges — weight lifting, cylinder stacking, and a walking race — demand strength, mobility, and dexterity. Creating a single versatile suit that can complete all three challenges is likely out of the scope of many garage tinkerers, so the actual Tetsujin competition and our scaled-down version of it allow different suits for each challenge. Last time, we began with an exosuit for the cylinder stacking challenge.

The Story Thus Far ...

Our cylinder stacking suit relies on a kind of scissor action to grip the cylinders (as a simple mechanism for manipulation) and a turntable to relocate the stack (for better balance while turning, and to avoid the dangers of having the person inside the suit hurt themselves while turning).

All of the additional mechanisms are powered by servos we pirated from FIRST Edurobot kits lying around Robot Central (our garage), and they are conveniently wired directly into the board of the Robonova. While the actual strength augmenting ability of this suit remains to be seen, it still models ideas on a small scale that could viably be used in the full scale Tetsujin competition, and perhaps beyond.

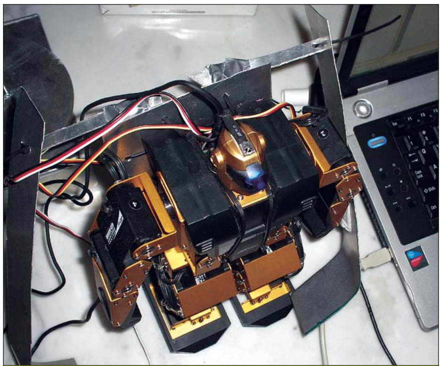
Not a Leg to Stand On

One of the first modifications on our list was to fabricate leg braces. One of the main ideas of the leg braces was to ensure that the Robonova balanced on the turntable. Balance would be achieved by having the leg braces be firmly attached to the Robonova as well as the turntable, making the entire exosuit a single unit. The braces would also discourage any unwanted motion on behalf of the Robonova itself.

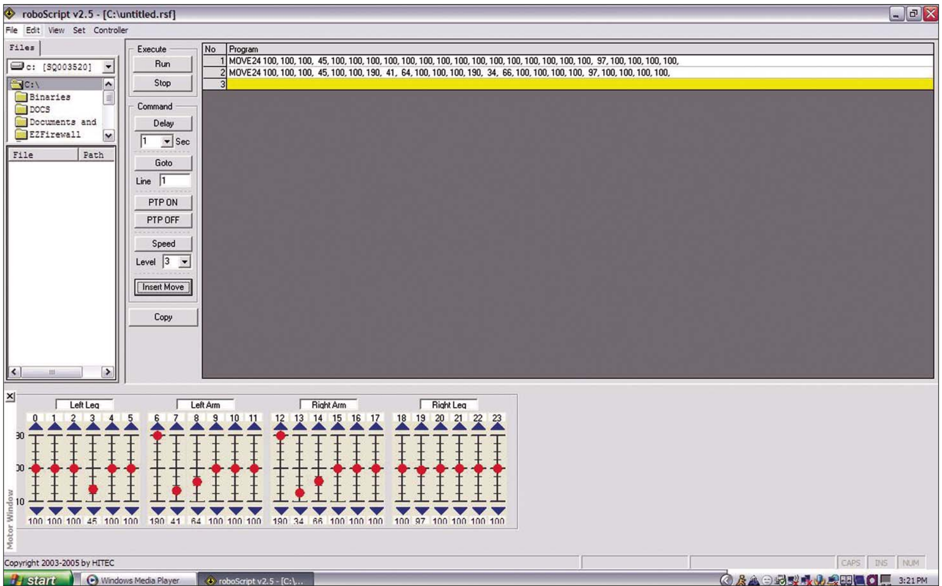
Later modifications could include some kind of way to allow an up and down vertical motion within the leg braces in case the Robonova needs to lower itself to grasp the cylinders (or raise itself to place them), but getting the Robonova to balance in the first place is most important. So we fabricated some unostentatious but functional leg braces out of some scrap aluminum in Robot Central.

Sometimes Robot Central is not, in fact, completely dedicated to

THE EXOSUIT!



ROBONOVA PROGRAMMING.



ROBOSCRIPT.

robots and, as a consequence of that, some of the pieces that we fabricated for the Super Robonova (namely our leg braces) ended up on a solar powered boat (don't ask). Only a minor setback, because we had plenty more scrap aluminum out of which to remake the parts.

The leg braces would serve the balancing function both when the Robonova is on and off, because the limp deactivated bot is quite a bother to work with. Now with the supportive braces, Robonova was easier to use as a model for further modifications without having to turn it on or reset the neutral position every time we wanted to try something in a new position.

Other modifications that needed to be made included shaping the end effectors. The arm extensions needed to succeed where the Robonova's plastic fists (and human hands, in real world applications) fail. Since the exosuit could be fashioned specifically for each task, the end effectors for the cylinder stacking challenge needed simply to conform to the cylinders that required manipulation.

Instead of tackling the ambitious complexity of mechanical fingers, we opted to fashion some simple curved end effectors that would conform to the sides of the cylinders and hold on to them via friction and pressure. The servo-powered arms of the Robonova would supply the pressure, and some rubber lining on the arm extensions would provide sufficient friction.

Going Off RoboScript

With the mechanical aspect of the hack essentially complete, we were ready to program the Robonova. The Robonova comes with two methods of programming: text-based RoboBASIC and a graphical interface called RoboScript. For the purposes of programming for the Tetsujin challenges, RoboScript proved to be a more efficient method of programming. Events like cylinder stacking and weight lifting involve relatively simple motions, so the simplified programming was a nice fit. The walking race demanded the complex motion of walking, but thankfully the Robonova came with demo programs that took care of this more difficult gait.

RoboScript conveniently labels the groups of servos by their respective appendages — right arm, left leg, etc. — which makes for user-friendly programming. Each group contains adjustable dials that can be manipulated to achieve the desired position with the Robonova. An adjustment of the dial corresponds to a proportional movement in the servo.

Perhaps the best accommodation that RoboScript offers is that if you have the Robonova hooked up to the computer while you are programming, adjusting a dial in the program will generate the corresponding motion in the actual Robonova; you will see the movements as you program them. This helps to take a great deal of the guesswork out of programming, because it allows the user to base their commands on qualitative observations of the Robonova and not just a series of abstract numbers.

The most significant limitation of RoboScript is that what was just described is essentially the extent of the program. It consists entirely of programming individual "moves" with the graphical dials. It does not have the capacity for subroutines or loops, let alone sensory input, so it is really best suited to simple programs that do not include repetitive sequences of motion. The essential motion of the cylinder

PRINCE MYSHKIN SYNDROME.

CYLINDER STACKING.

stacking is indeed simple; just grabbing, lifting, turning, putting down, and releasing. But this motion does need to be completed several times — once for each cylinder. This would be a prime location for a FOR loop in the RoboBASIC, but copy and paste should also do the job in RoboScript.

Stack Attack

With the mechanical augmentations complete and the programming done, it was time to test the Robonova's new exosuit for the cylinder stacking challenge. However, our efforts at testing were stymied by another attack of the Prince Myshkin syndrome mentioned in the previous article on the "Super Robonova" — it was essentially incapacitated by mysterious bouts of uncontrollable motion.

The Robonova's left leg would constantly kick forward, despite attempts to remedy the problem with software by resetting the zero point and with hardware by resetting the servo horn to start the leg at a position farther back. And when the Robonova was kept on for an extended period of time while trying to write programs, it would eventually have the type of mechanical seizure that earned it the literary moniker. So, although we did not have a competition deadline to complete the project by — like entrants in the Tetsujin competition do — we did have a deadline from our editor to finish this project, so we effectively ran out of time to execute more thorough troubleshooting and diagnostics on the ailing system.

Even so, partial tests showed the Robonova able to complete parts of the challenge, like gripping a cylinder. We're quite sure clever tinkerers could come up with some solutions to these mysterious problems, and we're also pretty sure these problems were out of the ordinary. But even though the effectiveness of our design could not be tested through the challenge, the process and implications of this project can still be evaluated. So what does an exosuit for the Robonova tell us about exosuits applied on a larger scale?

Microcosm in a Microcosm

Even though the Robonova provided a nice opportunity to model ideas for Tetsujin exosuit designs, it was still a very simplified model because many of the complexities of the full scale competition were eliminated by working with the tiny bot.

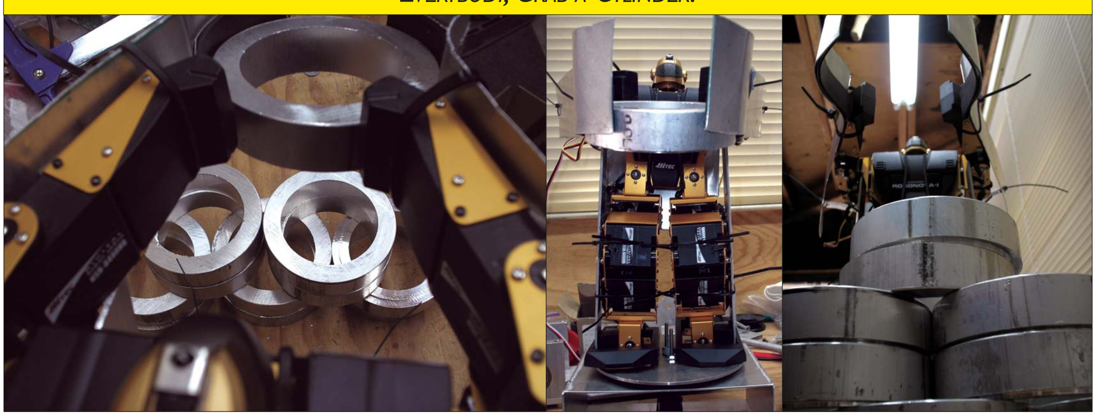
Perhaps the most obvious simplification achieved with the Robonova was in the area of safety. If the powered exosuit went crazy on Robonova, you might have to suffer the tragedy of buying a new servo, but if a full scale suit went berserk with a person inside you may be seriously dealing with a life and death situation.

A suit with the capability to lift in excess of 1,000 pounds certainly has the power to do some serious damage to a human being, especially considering the human operator is inside the suit itself. This means that just a single joint rotating too far could have disastrous consequences for the operator, unlike a separate mechanical unit that would actually have to attack the operator in some fashion to inflict comparable damage.

Carefully choosing a power source is another dilemma faced by Tetsujin competitors that we didn't have to worry about with the Robonova. With the Robonova we were able to wire the servos directly into the bot itself — a shortcut that would be akin to powering a mechanical exosuit by somehow hooking it up to the operator's heart or brain. Now, unless the Department of Defense has some crazy project up its sleeve for a future cyborg army, that kind of operation seems out of the realm of practical application. That means exosuit builders must consider how to carry or access a power source, whether it be batteries, a hydraulic pump, or an air compressor.

Any of these options come with a host of design considerations and difficult decisions —

EVERYBODY, GRAB A CYLINDER!



should the suit carry a generator of some sort (or compressor or pump) to make it self sustaining, or should it just carry reserve units of power (batteries or tanks)? Either option would include the issues of placement on the suit — perhaps in a backpack for walking suits, or maybe inside turntable assemblies for cylinder stackers. Or if the task was quite local to a small area, there may be the option of utilizing an off board power source, like a battery, compressor, or pump connected by cords or tubes.

These issues of safety and power sources imply an overarching difficulty with the overall size of full-scale exosuits. The truth is that they are indeed big machines, and big machines need high safety standards, lots of power, and also a lot of time and effort during construction.

The Robonova's exosuit was able to be made out of scrap aluminum from the garage because of the assumption that none of the tasks would demand a much stronger material. The same assumption cannot be made for full scale Tetsujin exosuits, so much more attention must be paid to materials selection and the like. Basically, a full-scale exosuit is quite literally a big challenge.

Finally, the human body is a far more complex template to build around than the mechanical body of the Robonova. A quick comparison gives a good idea of the increased complexity when building around a person as opposed to a bipedal servo walker: the Robonova has only 16 degrees of freedom, while the human body can be considered to have degrees of freedom ranging from the hundreds to the hundreds of thousands. That may seem like an excessive number of degrees of freedom, but the human body does indeed have a phenomenal range of motion, especially when forms of motion such as abduction (moving away), adduction (moving towards), flexion (bending), extension (stretching), circumduction (turning around), and rotation (rotation around an axis) are considered for each applicable joint. Of course, an exosuit does not need to be built around every degree of freedom, but the range of motion of the operator is certainly worth consideration.

Considerations generated by the degrees of freedom of the operator of an exosuit most notably include design constraints. In the interests of safety, efficiency, and effectiveness, the degrees of freedom of the human body unused in the exosuit should be constrained by the design of the exosuit. For example, for the cylinder stacking suit we modeled on the Robonova, only three degrees of freedom were required (one for the turntable, one for the scissor grip, and one for bending over to pick up cylinders). The unnecessary degrees of freedom were constrained by pieces like the arm extensions and leg braces. These constraints were necessary to ensure that only the desired motion was achievable by the suit.

The importance of constraints in exosuit design is even more evident when the full-scale case is considered. Unconstrained degrees of freedom in a Tetsujin suit would likely correspond to joints of the human operator that do not have an accompanying mechanical joint in the exosuit. For instance, if the cylinder stacking suit modeled by the Robonova was made for the full scale competition, one particular degree of freedom that would need to be constrained for safety reasons would be in the waist of the operator. The turntable is intended to do the turning, but if the waist of the operator was somehow unconstrained in the suit, there is the possibility that a dangerous load could be applied to the operator at the waist, possibly resulting in serious injury.

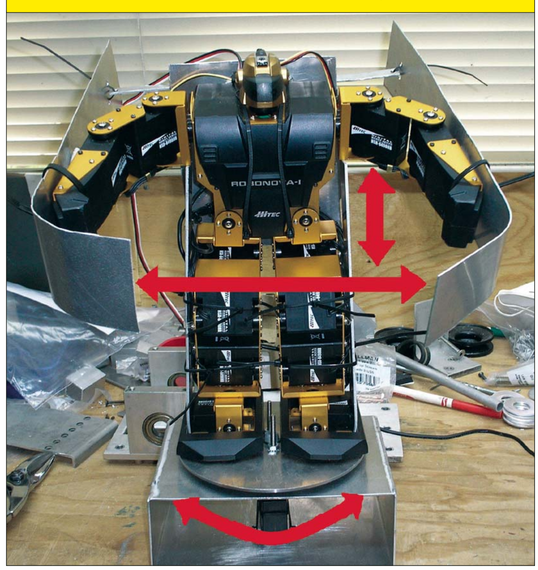
To constrain this degree of freedom in the operator, they should be somehow strapped into the suit to make it so that the exosuit itself will be the only thing making the risky motion. Of course, the operator should not be constrained to the point that it is impossible to operate the suit or escape the suit if anything goes awry, but designing the suit around the body of the operator in such a way as to constrain many unnecessary degrees of freedom should result in better safety, efficiency, and effectiveness of the design. Effective constraint design, however, demands careful attention to the subtleties of the human motion and could be a potentially quite difficult task.

The Tetsujin competition itself, though, is even a simplification of the challenge of building exosuits for the real world. Real world exosuits will have to function in an unpredictable environment full of extraneous variables and diverse goals, much different than the controlled environment the competition offers.

Commercial exosuits might be expected to lift uneven and irregular loads, unlike the balanced load of a weighted barbell. Real world exosuits might also be expected to grapple with a range of objects, from regular cylinders to large boxes to irregular boulders. They could even be expected to walk downhill or uphill instead of on flat terrain. This is not meant to diminish the difficulty or grandeur of the Tetsujin competition, but it is the simple truth that designing for the real world will be more complex than designing for competition.

This seems to beg the question, though — why even bother with

DEGREES OF FREEDOM.



mechanical exosuits if they are so complex? If you have to worry about all these things like safety and power sources and the subtlety of the human motion, why not just go with a forklift or an autonomous machine of some sort?

In truth, many possible applications for exosuits could be achieved just as effectively and perhaps more efficiently with existing technologies like forklifts. But there are still many applications that would benefit from the unique inclusion of the human element achieved by powered exosuit technology. For example, while a forklift might be suited to moving palettes of hardware around the Home Depot; it would not be very useful for a senior citizen with limited mobility that just needs help getting to the bocci ball field, or a soldier that could use help carrying heavy equipment on the uneven terrain of a battlefield.

Exosuits could also help those with disabilities more effectively than a wheelchair can, or they could perhaps help firefighters carry heavy lifesaving equipment into the heat of a fire. Applications like these require some kind of ability augmenting mechanical assistance that cannot really be effectively offered by existing technology.

Cutting Edge, Not Bleeding Edge

The bleeding edge generally refers to the point in an engineering design where increases in cost, even large increases, only result in miniscule increases in performance. Some might argue that development of powered exoskeletons is engineering at the bleeding edge of design. It could be said indeed exosuits could have practical applications, but at what cost? Are exosuits really the best answer?

The simple answer is a flat-out yes; exosuits are an answer to many applications like the ones listed above. The effectiveness of many of the solutions for the above listed applications is contingent on qualities that exosuits (or at least a future form of powered exosuits) are likely to display.

For example, increasing mobility for the elderly or impaired is something that is already achieved with technology like wheelchairs. Exosuits have distinct advantages over this existing technology. Some of the limitations of wheelchairs and similar mobility maximizing devices include their difficulty with terrain, cumbersome size, and lack of benefits in the area of recovery. Exosuits, once they reach a higher stage of development, have the possibility of offering the ability to cope with terrain by taking advantage of the human walking motion, a less cumbersome assistive machine by efficiently molding to the human form, and even some benefits in the area of physical therapy by exercising the muscles that need assistance in the first place.

Similar advantages can be listed for many applications for powered exosuits, so it is clear that exosuits can eventually serve very practical purposes. They may be a difficult answer to hard problems, but that's what progress is about — solving the hard problems to improve the quality of life. These applications — disaster response, battlefield assistance, increased mobility for the elderly or impaired — are all problems that could result in a significant improvement in the quality of life if solved. And exosuits certainly seem to be a logical solution, even if they are difficult to get right.

And because they are such a difficult technology to develop that is why events like the Tetsujin competition play such a vital role in development. Events like Tetsujin act as an intermediate bridge between the arena of ideas embodied by the Super Robonova and ambitious real world implementation. This isn't engineering at the bleeding edge; it's innovation at the forefront of progress. SV

Acknowledgements

We would like to extend a special thanks to Tony Ohm at Hitec for advice on the mysterious servo issues.

EVENTS CALENDAR

Send updates, new listings, corrections, complaints, and suggestions to: **steve@ncc.com** or FAX **972-404-0269**

If you're in Boston this July, make a point to see the AAAI Robot Competition and Exhibition. These folks always put on some unique and challenging events. This year, the events include The Robot Scavenger hunt in which robots must search the conference hotel for a list of items that include people and information. The items can only be found at specific locations and times. During their quest, the robots will have to deal with all the usual things found in the middle of a busy technical conference, including lots of people walking around. To succeed, a robot has to be good at interacting with people and the environment.

Human interaction is further emphasized by their next event, named (surprise!): Human Robot Interaction. In this one robots are scored on their ability to complete tasks that fall into several categories. The easiest examples include recognition of and reaction to human gestures, recognition of human emotion and appropriate emotional expression, and natural language understanding and action execution. Want a more challenging task? How about: shared attention, common workspace, intent detection. This task requires the robot to do things such as "remembering referents from previous sentences and being able to disambiguate 'this' and 'that'; following human eye gaze to determine objects of interest in the environment, and using shared attention in constructing referents in sentences or picking topics of conversation."

These sorts of events sound like a lot of fun to me and they promote the development of useful, general-purpose robots. I'm sure somebody out there is thinking, "if they just added a flame-thrower and power saw." You might prefer to check out the War-Bots Xtreme combat event in that case!

Know of any robot competitions I've missed? Is your local school or robot group planning a contest? Send an email to steve@ncc.com and tell me about it. Be sure to include the date and location of your contest. If you have a website with contest info, send along the URL, as well so we can tell everyone else about it.

For last-minute updates and changes, you can always find the most recent version of the Robot Competition FAQ at Robots.net: **http://robots.net/rcfaq.html**

— R. Steven Rainwater

July

7-10 **Botball National Tournament**
Norman, OK
Teams compete with autonomous robots built from standardized kits.
www.botball.org

11-13 **Singapore Inter-School Micromouse Competition**
Great World City, Republic of Singapore
Annual competition for student-built micromouse robots. Students from secondary schools, junior colleges, and technical schools have been participating in this contest for 16 years.
www.np.edu.sg/~adp-alpha/micromouse/mice_main.htm

16-20 **AAAI Mobile Robot Competition**
Boston, MA
This long-standing competition for autonomous robots typically includes the Robot Challenge, in which robots navigate the conference center; Robot Rescue, in which robots must locate injured humans in a disaster area; and Hors d'oeuvres anyone? in which robots must serve and interact with humans.
www.aaai.org/Conferences/National

17-21 **K'NEX K*bot World Championships**
Las Vegas, NV
Includes three events: Two-wheel drive K*bots (autonomous), Four-wheel drive K*bots (autonomous), and Cyber K*bot Division (R/C).
www.livingjungle.com

22-23 **War-Bots Xtreme**
Saskatoon, Saskatchewan, Canada
Radio-controlled vehicles destroy each other.
www.warbotsxtreme.com

27 **AUVS International Aerial Robotics Competition**
US Army Soldier Battle Lab, Fort Benning, GA
Flying robots are required to complete a fully autonomous ingress of 3 km to an urban area, locate a particular structure from among many, identify all of the true openings in the correct structure, fly in or send in a sensor that can find one of three targets, and relay video or still photographs back 3 km to the origin in under 15 minutes. And that's just one of three scenarios!
http://avdil.gtri.gatech.edu/AUVS/IARCLaunchPoint.html

ROBOTS.NET

August

2-6 **AUVS International Undersea Robotics Competition**
US Navy TRANSDEC, San Diego, CA
Autonomous underwater robots must complete a course with various requirements that change each year.
www.auvsi.org/competitions/water.cfm

4 **Elevator:2010 Climber Competition**
Mountain View, CA
Autonomous climber robot must ascend a 60 meter scale model of a space elevator using power from a 10 kW Xenon search light at the base.
www.elevator2010.org/site/competition Climber2005.html

20 **RoboCountry**
Takamtsu City, Kagawa, Japan
Described on the website as *humanoid robot combat* presented by the Kagawa Humanoid Robot Society.
www6.ocn.ne.jp/~robotics

September

9 **SWARC Texas Cup**
Mike's Hobby Shop, Carrolton, TX
Radio-controlled vehicles destroy each other Texas-style.
www.robotrebellion.net

Robotics Showcase

NEW PRODUCTS

COMMERCIAL ROBOTS

Pioneering Drinkmation™

Motoman's new RoboBar™ HP offers complete robotic bartending and beverage dispensing solutions. Three versions are now available: Robobar HP, Robobar E, and Robobar NA

RoboBar HP — Faster than the most experienced human barkeep, able to produce a perfectly mixed drink every 10-15 seconds, Motoman's new RoboBar™ high-production model is a complete, self-contained robotic bar designed for use in casinos and other high-volume service bar applications where human servers deliver the drinks to customers.

The bartender is a unique, dual-arm Motoman DA9IC robot with a compact NXC100 controller housed in its base. The two manipulator arms on this innovative robot each have five axes of motion, and the base also rotates to provide an eleventh axis of motion, allowing RoboBar to perform a wide range of operations quickly, accurately, and efficiently.

One robot arm is equipped with a simple parallel jaw gripper that handles cups, glasses, and beer bottles. Up to eight dispensing guns are mounted on the robot's other arm. Each gun can dispense up to 16 different ingredients (128 total), including liquors, mixes, juices, and wines — in any combination. RoboBar is not only fast, it mixes each drink perfectly every time, eliminating lost revenues due to spillage and overpours. The robot places multiple drinks onto a tray that shuttles in and out of the cell. A safety enclosure is included. The robot is highly reliable and programming it is simple. The user interface is intuitive and graphics-based. Servers enter their drink orders using a touch screen interface, which also identifies each drink on the tray. The number of drink recipes that can be programmed is virtually unlimited. Various options can be configured to customize RoboBar HP to meet the unique needs of specific service bar applications.

RoboBar E — For lower-volume applications, the RoboBar E (Entertainment model) is a "star pourer" that draws people like a magnet. This model uses the same dual-arm DA9IC robot equipped with simple parallel jaw grippers mounted on each arm that allows it to operate much like a human bartender — only better. RoboBar E is designed to use a magnetic card scanner to authorize drink service. After a valid card swipe, the customer uses a touch screen to choose a beverage. The Motoman robot selects a cup, and then fills it with the appropriate beverage(s) and ice, if desired. The robot holds the glass or cup in one gripper while it pours from a bottle held in its other gripper. The robot might also move the cup to a dispenser for ice, beer, wine, juices, or soda, as needed, before placing it on a drink delivery slide for customer pickup.

RoboBar E includes the robot, dispensers for beer, soda and juices, cups, and ice. A flat-panel video screen provides a selectable "personality" for your RoboBar. Customers can choose a male or female personality, with matching voice. The RoboBar personality can also be customized.

RoboBar NA — Operating much like the RoboBar E model, Motoman offers a RoboBar NA (No Alcohol) version designed to dispense hot coffee drinks, such as coffee, espressos, cappuccinos, and lattes, as well as a variety of soft drinks, such as sodas, juices, and other non-alcoholic beverages. However, since no proof of legal drinking age is required for non-alcoholic beverage purchases, RoboBar NA does not require use of a magnetic card scanner to authorize drink service.

The RoboBar E and NA models are available for purchase, lease, or event rental. The RoboBar HP model is available for purchase or lease only.

For further information, please contact:

Motoman, Inc.
805 Liberty Ln.
West Carrollton, OH 45449
Tel: **937•847•6200**
Email: **info@motoman.com**
Website: **www.motoman.com**

INDUSTRIAL ROBOTS

Combined Six-Axis and Linear Robot

KUKA Robotics Corporation — a leading global manufacturer of industrial robots — offers the KUKA

JET robot which is a six-axis robot mounted on a linear unit. The new robot is designed for customers with applications that entail long reach tasks. The six-axis robot is mounted either upside down or sideways on the linear rail, depending on the application and is available in four configurations with different reaches and working ranges.

"Customers with applications where long distances need to be traversed will find the KUKA JET robot ideal," said Kevin Kozuszek, director of marketing for KUKA Robotics. "The robot combines the speed of a linear axis and the flexibility of a six-axis robot making it ideal for handling tasks in multiple industries including injection molding, die casting, machine tool manufacturing, and logistics."

The KUKA JET robot's enhanced maneuverability allows machines to be tended through narrow openings and parts precisely positioned even within the machine. It also allows parts to be withdrawn from the machine in a longitudinal direction. This makes it possible to serve a number of machines in a row, resulting in optimal material flow. Up to two robots can be controlled on one linear axis. Additionally, the installation can be configured to allow several machines to be tended by one combination. Payloads range from 30 to 60 kilograms.

The company's five- and six-axis robots range from 3 kg to 570 kg payloads, and 635 mm to 3,700 mm reach, all controlled from a common PC based controller platform. KUKA robots are utilized in a diverse range of industries including the appliance, automotive, aerospace, consumer goods, logistics, food, pharmaceutical, medical, foundry, and plastics industries and in multiple applications including material handling, machine loading, assembly, packaging, palletizing, welding, bending, joining, and surface finishing.

For further information, please contact:

KUKA Robotics Corp.
22500 Key Dr.
Clinton Township, MI 48036
Tel: **586•569•2082** Fax: **866•329•5852**
Email: **info@kukarobotics.com**
Website: **www.kukarobotics.com**

MOTOR CONTROLLERS

Stepper Motor Controller

A new low-cost single axis stepper motor controller/driver from Techno-Isel is completely self-contained and comes ready to plug in for immediate use. It is capable of controlling and driving two or four phase stepper motors and features integrated I/O. This controller/driver is designed to perform a variety of automation-related, motion control, inspection, dispensing, and production applications.

The controller — identified as the Centurion — features as standard a 32K battery backed memory capable of storing 10 programs (switch selectable) and up to 5,000 motion commands, eight digital inputs, eight digital outputs, operator control panel, remote start and stop capability, manual jog feature, watchdog timer, and motion control software. The controller is designed to communicate with a PC via an RS232 interface. Connections for I/O are made with plug-able screw terminals located on the controller's back panel and motor connection is made with a nine-pin D connector.

The Techno motion control software included with the controller, is a user-friendly, integrated programming environment. It features a program editor, compiler, communications program, and jog program with teach mode. The editor and compiler allow the editing and compilation of a motion control program using simple, English-style commands. The communications program allows complete control of and transfer of a program from the PC to the controller. The jog program allows manual positioning of the motor from the PC's keyboard. It also has a teach mode which will automatically generate a program. Once a program is loaded into the controller's memory, it may be controlled either from the PC or from the controller's front panel. The Centurion controller may also be completely disconnected from the PC for completely stand-alone use.

The Centurion controller is available with a choice of three different motor drivers capable of providing continuous currents from 1-6 A per phase.

For further information, please contact:

Techno-Isel Linear Motion Systems
2101 Jericho Turnpike
New Hyde Park, NY 11040
Tel: **800•819•3366** Fax: **516•358•2576**

Show Us What You've Got!

Is your product innovative, less expensive, more functional, or just plain cool? If you have a new product that you would like us to run in our *New Products* section, please email a short description (300-500 words) and a photo of your product to:

newproducts@servomagazine.com

COMBAT ZONE

Featured This Month

Participation

28 *Organizing a Combat Event* by Kevin Berry

30 *Safety Tip — Installing Holes* by Kevin Berry

Feature

30 *Battle Beach 4 Rocks the South* by Kevin Berry

Events

31 *Upcoming — July*

33 *Results — April and May*

Technical Knowledge

32 *Tips for Combat Robot Builders* by Steve Judd

Product Review

33 *Vantec RDFR23 Speed Controller* by Kevin Berry

PARTICIPATION

Organizing a Combat Event — Not For the Faint of Heart

● by Kevin Berry

You just left your first combat event and you are totally jazzed. The people, the destruction, the pure fun have you hooked. Like many new builders, you want to share the fun back home at your school, club, or community. Before you rush into holding your first event, you need to think about what's required, and avoid some of the pitfalls new (and experienced) organizers stumble into.

First, you need to make a key decision, following something known in the combat community as "Judd's Law." Per this piece of hard-won wisdom, you need to decide if you're holding a show or an event. By a commonly-accepted definition, in a "Show," the fights happen per a rigid schedule, and if someone isn't ready, they forfeit. In fact, at many shows, there is no actual competition outside the box — no winners or losers. This format maximizes excitement and predictability for the spectators and venue, while setting aside the unpredictable nature of combat damage, and the variable time needed to repair broken bots. This allows selling tickets for specifically scheduled sessions.

An "Event," on the other hand, revolves around the builders, and spectators view on a "catch as catch can" basis. Sets of brackets, while tentatively scheduled, are run only when ready, and fights may be postponed if the bots aren't repaired. This isn't to say that Event Organizers (EOs) don't try to keep things moving, or have alternate brackets planned. It's just that things are a bit looser — schedule-wise — and priority rests with the builders.

So, you've decided on your format, and its time to put in place the "Three Ps" of an EO: People, Promotion, and Process. The combat community is very

forgiving of people who try hard and seek advice and assistance, but very critical of those who ignore good advice, and even tougher on those who try to "ego" their way through an obvious failure. So your approach should be a humble heart, a steely will, and lots of communication.

People

The "people" part of the equation is critical. Many, if not most, disastrous event failures can be traced to lack of dedicated bodies to plan, set up, run, and tear down an event. While builders are VERY willing to pitch in, they really ought to be spending their time honing bots and charging batteries, rather than building an arena at the last second, or helping construct brackets. Assembling a dedicated staff ahead of time is one key to ensuring success. If the staff actually knows what they are doing and are given freedom to manage their areas, so much the better. But even having a table with a sign marked "Registration" with an inexperienced volunteer, who can take money and weigh bots, is a step above what some events wind up providing.

Promotion

Promotion is a very tough subject. Some EOs have learned — to their chagrin — that just scheduling an event and trusting that enough people will show up is a quick route to failure. On the other hand, starry-eyed expectations of rowdy crowds, knocking down barricades in their rush to buy expensive tickets haven't typically materialized. The news media has been ho-hum about the whole sport since TV coverage died, although many local papers will run articles if there is a "human interest" angle.

Hard-won experience teaches that, with a few notable exceptions, it's easier to take the event to the crowds, rather than drawing masses of people to a remote venue. Insect events have done well at shopping malls, school fairs, college open houses, and hobby stores. Bigger events have drawn crowds when paired with other large gatherings, like Labor Day's Dragon*Con in Atlanta, Combots shows in California, or ROBOlympics in San Francisco.

The second part of promotion is getting builders to come. The Delphi Forums, Builders Database, and RFL site are common places to advertise. In my experience, personal contact is the single best way to draw builders. For smaller, 20–30 bot insect events, I usually send 150–200 emails over the three months prior to the event, plus uncounted IM chats. This includes known builders, and "blind" emails to school districts, clubs, charitable organizations, clubs, and universities.

When I organized the insect portion of Battle Beach 2 — by actual count — I sent over 200 emails to builders, and can't even begin to estimate the number of telephone calls and IM chats.

Process

The process of organizing any event — whether it's a charity walk-a-thon or combat robot fight — is simple but critical. Team members have to know what their responsibilities are, how much authority they have to make decisions, and when they are to complete tasks. Leading an all-volunteer team is high art, requiring a unique set of skills.

There are a couple of components to the "mechanical" side of event organization, those being: have a written plan and communicate intensely. If you, as a new EO, write down your "to-do" list, solicit similar lists from veterans, and get your team (and outsiders) to review your planning, you'll do well. It's amazing how many obvious things get forgotten when you're working without a plan.

Your plan should be in place AT LEAST three months before the event. The bigger the event, the farther ahead you should have your list ready (up to six or more months for large gatherings).

Communicate, communicate, communicate. Answer emails and phone calls until you're sick of them, then do it again. Opinions on this may vary, but I'd plan to spend four to 10 hours a week just communicating between your team members, key builders, sponsors, venue providers, and others. Besides the 200 emails to builders I mentioned above, I know I've sent and received well over 500 between the key three to five team members who are going to run an event. This point cannot be

EVENT ORGANIZER'S CHECKLIST

- ❑ Venue commitment and rules
- ❑ Arena(s) and setup/teardown crew
- ❑ Tent or area for pits, power strips, extension cords, tables, chairs
- ❑ Dirty Work area and tools
- ❑ Safety official
- ❑ Announcer, PA/boom box and music
- ❑ Referees, judges
- ❑ Frequency clips
- ❑ Timer, tap out lights
- ❑ Scales, check-in volunteer
- ❑ Publicly stated expectations of sportsmanship, fun, tough but fair fights, enforcement of safety
- ❑ Rule set
- ❑ Coordination on transportation/storage for the arena before and after the event
- ❑ Access to a large pool of experienced builders/fighters
- ❑ Crowd control devices and/or traffic directors
- ❑ Parking, loading zones
- ❑ Awards, prizes, publicity
- ❑ Sponsors
- ❑ Insurance, Fire Marshall, Public Safety
- ❑ First Aid kit, fire extinguisher
- ❑ Treasurer, entry fees, payments
- ❑ Lighting, signs, staff tables, and chairs
- ❑ Board and/or computer for brackets
- ❑ RFL sanctioning
- ❑ Builders Database

stressed enough — nothing kills an event quicker than poor or non-existent communication!

Pain

Finally, here's the last key to running an event. (I know I said there's only three, but consider this a bonus item.) Repeat EOs have to have the thickest skin on any mammal, bar none. Survival of the fittest matters here, too. There will ALWAYS be someone who could have done it better, thinks you're a jerk, or feels unfairly treated. If you've followed the steps above, though, you know you've satisfied the majority of the community, and your reputation will grow. Nothing beats the satisfaction of hearing from happy builders that they can't wait for your next event.

So, if you're excited about the sport and want to hold your own event, press ahead! Just remember the key ideas I've explained, and get ready for the headaches, stress, and incredible satisfaction headed your way. **SV**

Safety Tip — Installing Holes: Drill Safety for the Home Hacker

● by Kevin Berry

Marc DeVidts, creator of the Builders Database, shows how to break all three rules, plus a couple more! Photo courtesy of Marc DeVidts.

Drilling holes is one of the most basic shop tasks and also one of the most dangerous. There are dozens of mistakes that can be made while drilling, and I've done them all.

Rule #1 is to wear safety glasses. Everybody thinks they can do "just this one hole" without them, but getting a shaving out of your eye soon teaches the value of wearing them every single time.

Rule #2 is to always use clamps to hold the material — whether using a handheld drill or a press. Invariably, either the drill binds and the material spins, gouging soft tissue along the way, or else the drill punches through into whatever is holding it (often the hand of the driller). The motor spinning the drill is much stronger than the human hand, otherwise we'd all just hold the bits in our fingers!

Rule #3 on the hit parade is to properly size the drill bits to the job. Drilling progressively larger holes is often safer than trying to hog out a giant hole all at once. Also, a common mistake when using small diameter bits is having too much outside the chuck, causing them to snap easily. Refer to Rule #1 about this one. Home builders are used to thinking about drilling as a mundane task, but true machinists will spend ten times as long setting up to drill a hole, as they do actually spinning the bit. Safety and precision go hand-in-hand, but the material doesn't!

In the frenzy of last minute building, or repairing in the pits, it's easy to say "this one time won't matter." Well, there's a saying my shop teacher used on us — "there's never time to do it safely, but there's always time for first aid." Blood is your friend, as long as it's on the inside. Keep it there! **SV**

BATTLE BEACH 4 ROCKS THE SOUTH

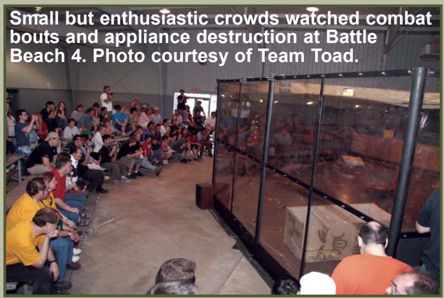
● by Kevin Berry

Battle Beach 4 was held on April 8th and 9th, at the Volusia County Fairgrounds in Deland, FL. About 60 bots — ranging from 150 gram Fairyweights to 220 pound Heavyweights — slugged it out in the two arenas. This was a builder-oriented event, heavy on action. Several classes ran round-robin formats, allowing more fights per team than a standard double elimination tree. Especially popular with the spectators were appliance demolition exhibitions and ad hoc rumbles.

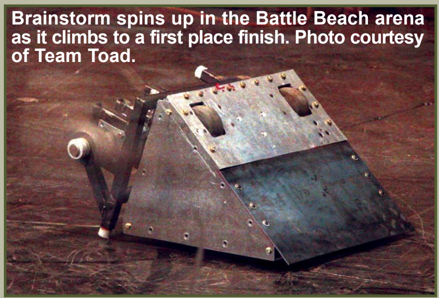
The new venue was much appreciated by builders, with air-conditioned pits and a hard roof to fend off the traditional Battle Beach monsoon rainstorms. While the rains held off this year, having both pits and arenas in one building was a much-appreciated perk to builders who came from as far away as California, Oregon, New York, Pennsylvania, and Wisconsin to enjoy the action.

Battle Beach's sponsors included Vantec, 80/20 Surplus, Titanium Joe, Microbotparts, UI Productions, Team Ninja, Team Moon, *Robot Magazine*, and, of course, *SERVO Magazine*.

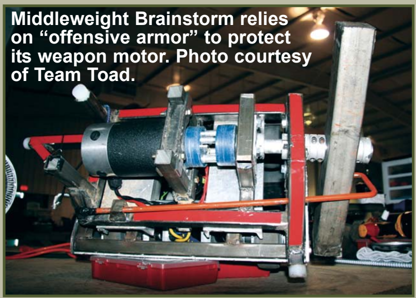
CM Robotics' HW Ziggy shows why wheels, motors, and a weapon are requirements for a combat bot! Photo courtesy of Team Moon.



Small but enthusiastic crowds watched combat bouts and appliance destruction at Battle Beach 4. Photo courtesy of Team Toad.



Brainstorm spins up in the Battle Beach arena as it climbs to a first place finish. Photo courtesy of Team Toad.



Middleweight Brainstorm relies on "offensive armor" to protect its weapon motor. Photo courtesy of Team Toad.

Fight Results

Fairyweights — 1st: Doodlebug, Team Ninja, Pusher; 2nd: Puckthud, Team Thorn, Thwack; 3rd: Puckpump, Team Thorn, Horizontal Spinner.

Antweights — 1st: Peligo, Berserk Robotics, Horizontal Spinner; 2nd: Pirhana, Team Ninja, Vertical Spinner; 3rd: Pop Quiz, Team Test Bot, Horizontal Spinner.

Beetleweights — 1st: John Henry, Legendary Robotics, Wedge; 2nd: Ron, Overvolted Robots, Saw/clamper; 3rd: Creepy Crawler, Team-X-Bots, Wedge.

Mantisweights — 1st: Mantis From Hell, Team V. Wedge; 2nd: Thwaxident, Insane Robotics, Thwack; 3rd: Rhino Viper, Team Diamond Back, Horizontal Spinner.

Hobbyweights — 1st: Flight Risk, Team Shenanigans, Horizontal Spinner (gasoline); 2nd: KITT, Team Moon, Wedge; 3rd: Test Bot, Team Test Bot, Lifter.

Featherweights — 1st: Totally Offensive, Team Mad Overlord, Horizontal Blade (currently ranked #1 in RFL); 2nd: Eat Hitch and Die, Team Skarn, Pusher; 3rd: Poetic Justice, A.G. Robotics, Wedge.

Totally Offensive — a perennial national level contender — racked up another win at Battle Beach.

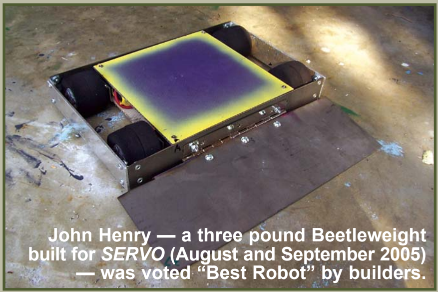
Lightweights — 1st: Ground Zero, Team O-Town, Full Body Spinner; 2nd: Crocbot, Team Gator, Speed Bump; 3rd: Street Thug, Skarn, Beater.

Middleweights — 1st: BrainStorm, Big Bang Robotics, Horizontal Spinner; 2nd: Lionheart, Team Toad, Wedge; 3rd: Ice Cube, Team Toad, Wedge.

Heavyweights — 1st: Eugene, Team Moon, Horizontal Spinner; 2nd: Pandemic, Weapons of Miniature Destruction, Vertical Spinner.

Other Awards
(By Builder, Judge, and Audience Vote)

- Best Battle Beach Rookie — Ziggy, CM Robotics
- Judge's Award — Pandemic, Weapons of Miniature Destruction
- Best Driver — Fuzzy Maudlin, Team Toad
- Sportsmanship Award — Alex Grant, A.G. Robotics
- Best Engineered Robot — Pandemic, Weapons of Miniature Destruction
- Best-Dressed Team — Team Toad
- Best Robot — John Henry, Legendary Robotic
- Most Awesome Loss — Household Trash, Divine Mechanics SV



John Henry — a three pound Beetleweight built for *SERVO* (August and September 2005) — was voted "Best Robot" by builders.

EVENTS

UPCOMING — July
(all RFL National qualifiers)

War-Bots Xtreme — WBX-III, Saskatoon, Saskatchewan, Canada. 7/22–7/23 2006. **www.warbotsxtreme.com** This is Canada's BIGGEST combat robot tournament. The weight divisions range from Ants to Heavyweights, PLUS Bot Hockey.

WBX-III

Team Think Tank — SNF Qualifier, Pasadena, CA. 7/22/2006. **www.teamthinktank.com** Included at this event are Fairy, Ant, and Beetle Weight classes. SV

SNF Qualifier

TECHNICAL KNOWLEDGE

Tips for Combat Robot Builders

● by Steve Judd, Team Tentacle

Getting Started

● *Read a book or two*. In particular, I recommend *Robot Builder's Bonanza* by Gordon McComb and *Kickin' Bot* by Grant Imahara.

● *Visit some builder's websites* — Steven Nelson's **http://teamkiss.com**, Ted Zeiger and Pete Covert's **http://teamcosmos.com**, and my own **http://architeuthis-dux.net** are all good starting points.

● *Start small*. Build a 1 lb or 3 lb robot first. A small robot can be built with inexpensive, readily-available parts. Small radio-controlled toys make an excellent platform for a first robot.

● *Don't rule out non-combat robotic sports* like FIRST Lego League, FIRST, BotBall, and others. These offer well-organized competitions (usually for school-sponsored teams) where you can gain a wealth of robotic experience.

Spending Your Bot Money

● *You get what you pay for*. There is a fine line between "inexpensive" and cheap.

● *Know what you're buying and know why you are buying it*. Think before spending your money — can you afford to buy a replacement if the item you are considering does not work out?

● *Don't skimp on your radio or speed controllers* — these are the most crucial parts of your bot. A good radio can be used for years, as can good ESCs.

● *Buy the best quality tools you can afford*. Some quite good tools can be had at very reasonable prices from discount suppliers, but if you can afford better — BUY IT.

● *Industrial surplus is your friend*. You can get a lot of quality bot components from industrial surplus dealers, manufacturers' surplus sales outlets, etc. Keep Point 2 in mind while shopping a surplus dealer.

Designing and Building

● *Do the most complete design you can*. CAD software is an effective tool if you have it or can get it. "Cardboard-aided Design" is a cheap and effective alternative — cut the pieces of your bot out of cardboard and fit them together. The more you know about how your bot will be assembled, the easier the fabrication will be.

● *Keep your design as simple as it can be*. This does not mean to build only simple bots — it means that you should not add anything to a design that is not there to make it stronger, faster, or better in some definite way. A well-executed simple design is often a lot cooler than a design so complicated that it's hard to execute.

● *Neatness counts!* You don't score any match points for this, but a clean, well-organized interior and an exterior with good fit and finish will help you in the arena, and get you some "style points" in the form of admiration by your fellow builders.

● *Design a bot that is easy to repair* — you will often need to make repairs in a hurry.

● *Remember to allow for the wiring harness*. The wiring inside a bot always seems to take up a LOT more space than you think.

● *Don't use sheet metal screws, pop rivets, and the like for assembly* — use quality bolts and machine screws. If your bot has a frame, weld the frame members together.

● *Standardize on a single fastener size, if possible*. Fewer different sizes means easier repairs.

● *Set screws are bad news*. Rotating parts should be secured to shafts with keys or keyless bushings (i.e., TranTorque, Shaftloc, etc.).

● *Pins are almost as bad as setscrews*. If your bot is dynamically stable in some position, you will end up in that position no matter how unlikely it seems. Design your bot so you can get back on your wheels from any orientation.

● *Any electrical connection that can come loose will*. All electrical connections need to be positively secured. Friction fits that "feel tight enough" are not.

● *Thread-locking products are your friends*. If a bolt or machine screw is intended to be "permanent," fix it in place using an appropriate thread-locking product.

● *Design your bot so changing radio frequencies is as fast and easy as possible*. You will often be called upon to switch to a different frequency at the last minute.

Competing

● *Driving: practice, practice, practice*. The bot needs to become

an extension of your body. Matches are won and lost when a driver looks away from the bots for a split second.

● *Show up on time with your bot complete and ready to run*. Passing safety on the first try should be the norm — not an exception. Arriving early and passing safety well before the start of combat gives you time to relax and socialize with the rest of the competitors.

● *After a fight, IMMEDIATELY service the robot*. Just because it looks fine on the outside does not mean everything is fine on the inside. The time to find this out is right after your previous fight, not just before your next one, or worse, in the arena.

● *Have at least two sets of batteries* — more if possible. You may not have time to fully recharge batteries between fights.

● *Bring spares for everything*. Since this can add a lot to the cost, standardize wherever possible on parts commonly used by other builders. This will make it easier to get an emergency replacement part if you run out of spares.

● *Keep your pit area clean and well organized*. You don't want to be searching for a critical part or tool for 10 minutes when you only have 20 minutes between fights.

● *Label your tools*. All major brand power tools look alike.

● *Be civil*. The other competitors are your best resource at any competition. Most will cheerfully lend you tools, give advice and assistance, and do whatever else is in their power to help you out if you ask nicely. Just remember that everyone is under pressure — just like you are — and might be busy with their own robots.

● *Pay close attention to the event staff and treat them with respect*. They are under a lot of pressure, too.

● *Be gracious whether you win or lose*. SV

PRODUCT REVIEW — Vantec RDFR23 Speed Controller

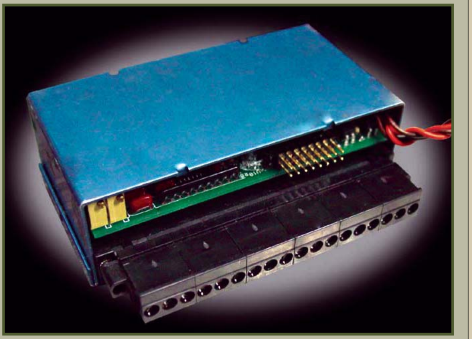
● by Kevin Berry

Vantec's series of speed controllers are battle proven, and loved by many top builders. They also supply to law enforcement, fire department, and surveillance applications. Their RDFR23 model supplies up to 30 amps continuous DC current at 30 volts. It can also handle a 70 amp startup surge.

There are a variety of mixing options, from straight "tank steer" to linear, to special exponential curves. There is also a special mix just for marine applications. The RDFR23 also provides dynamic braking, which is very nice for combat applications, where motors are constantly slammed from forward to reverse and back.

From a durability standpoint, the community generally agrees this unit meets their toughness criteria. At the Robocide event, Lighweight Chupacabra took a fearsome beating from top spinner "2EZ," with virtually every component being sliced. The RDFR23, despite a direct hit, dented case and with all the wires ripped loose, worked perfectly when re-installed.

The folks at Vantec are big supporters of the sport, donating prizes at many major events. SV



The RDFR23 measures 4.25" x 2.7" x 1.3" and weighs nine ounces.

EVENTS

RESULTS — April and May

Maker Faire: April 22, **www.combots.net** — World-class bots battled it out in this exhibition event, so there were no winners or losers. Fun was had by all!

Central Illinois Bot Brawl, May 6 **http://circ.mtco.com** — A small but enthusiastic set of bots fought at the Lakeview Museum, with teams from Illinois, Indiana, and Ohio competing. Solarbotics provided sponsorship, with assistance from Parallax, HobbytownUSA, and Lynxmotion. Results are as follows:

● *RFL Qualifier, 1 lb Combat:* 1st: Skeletor, P_Robotics, Evan Gandola; 2nd: Killer Aluminum Sandwich, Iron Fist, Rob Harnden II; 3rd: Evil Doorstop III, P_Robotics, Evan Gandola

● *500 g Sumo:* 1st: Orthos, dbots, Mike Dvorsky; 2nd: Sumo04, Black Bots, Andrew Black; 3rd: LYBOW, Black Bots, Andrew Black

● *LEGO Sumo:* 1st: Junior, dbots, Matthew Dvorsky; 2nd: Mighty Man, Brooksbots, Rick Brooks; 3rd: Net_Op_School, P_Robotics, Evan Gandola

● *1 kg Sumo:* 1st: Extrasensory, Brooksbots, Rick Brooks; 2nd: Exhume, Brooksbots, Rick Brooks; 3rd: BJ, dbots, Mike Dvorsky

● *3 kg Sumo:* 1st: Cheeky-san, dbots, Mike Dvorsky; 2nd: Excuse, Brooksbots, Rick Brooks; 3rd: Executioner, Brooksbots, Rick Brooks

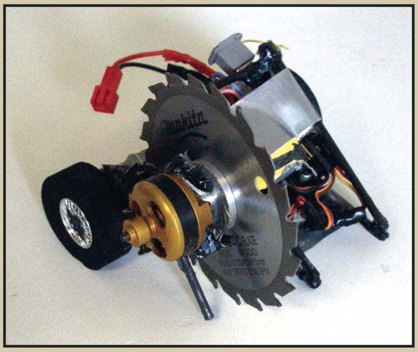

LEGO Sumo bots Junior and Mighty Man mix it up on the dojo at CIRC's Bot Brawl. Photo courtesy of CIRC.

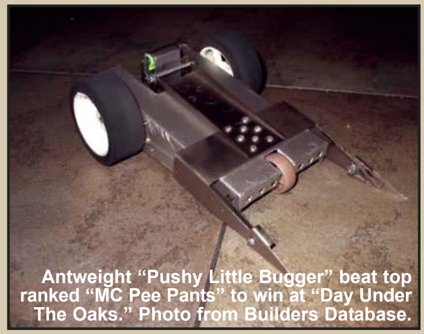
SRJC Day Under The Oaks: May 7 — Fifteen insect bots competed in a friendly, outdoors meet. Results are as follows:

● *Fairyweights:* 1st: VD, Team Fatcats, Vertical Blade (this team is currently ranked #1 in RFL); 2nd: Micro Drive, Team Misfit, Lifter; 3rd: Baby Bunny, Team Misfit, Wedge; 4th: Crisp, Offbeat Robotics.

● *Antweights:* 1st: Pushy Little Bugger, Tinkers Guild, Lifter; 2nd: The Bomb, Team Misfit, Drum; 3rd: MC Pee Pants, Team Fatcats, Drum (currently ranked #1 in RFL); 4th: Honey Bunny, Team Misfit, Wedge. **SV**

Vertical spinner VD continues to dominate the RFL's Fairyweight class by scoring another first place win. Photo from Builders Database.




Antweight "Pushy Little Bugger" beat top ranked "MC Pee Pants" to win at "Day Under The Oaks." Photo from Builders Database.

Winners from the "Day Under The Oaks" competition (L to R: Andy Sauro, Terry Slocum, Zachary Lytle. Front: Danielle Donaldson).

Bio Feedback Continued from page 7

2002/Jan 2003 issue of the same magazine (it comes out once every two months) for some very important updates on safety and modifications to the original article (pages 3 and 35). Just keep it away from children. (By the way, I find their site a bit hard to navigate.)

16) For lubricant, try and buy some tapping oil. Sometimes you can get some for free at machine-tool shows and the like. Sometimes the places that sell tools will give you a small, free sample bottle. A small bottle can last a long time, especially if you only tap once in a while. For aluminum, I usually dilute the tapping oil with Varsol. In reality, any lubricant would be better than tapping completely dry — if you have to and you're desperate, use old motor oil.

17) Recycle! The next time you throw out a toothbrush, keep it to clean the threads of the tap. Sometimes you can break a tap because the chips are not cleared out.

The readers may find some of the following formulas useful: In these formulas the following terms will be used: Nominal Diameter (ND): This is the outside diameter of an external thread (also known as the Major Diameter), for instance the nominal diameter of a 1/2" bolt is 1/2" (.500").

Thread Pitch: (P) the distance between the crests of two consecutive threads (the distance from the crest of one thread to the crest of the next thread), measured along the length of the thread. Most inch threads are written in the form of: 3/8-16 where 3/8 is the nominal diameter (outside dia.) (.375"), followed by the pitch, expressed as Threads Per Inch (TPI). In this case of 3/8-16, there are 16 threads per inch, therefore one inch divided by 16 threads results in a distance between two consecutive thread crests of 1/16 = .0625". To use another example, a 3/4-10 thread has a pitch of 10 threads per inch = 1/10 = .100".

Minor Diameter (MD): The diameter that is at the root (bottom) of the threads.

Thread Depth (TD): The distance from the outside of a thread to the bottom of a thread (a radius distance, useful for machining threads on a lathe).

Pitch Diameter (PD): The diameter that lies equidistant between the Nominal Diameter and the Minor Diameter.

All the following examples will use the threads of 5/16-18, for which ND = .3125" and the Pitch (P) = 1/18 = .0555".

Formula: Thread Depth (TD): TD = .6495 * P Example: .6495 * .0555 = .0360". Note 1: .6495 is a constant. Note 2: This value is a radius value and is useful for machining threads on a lathe.

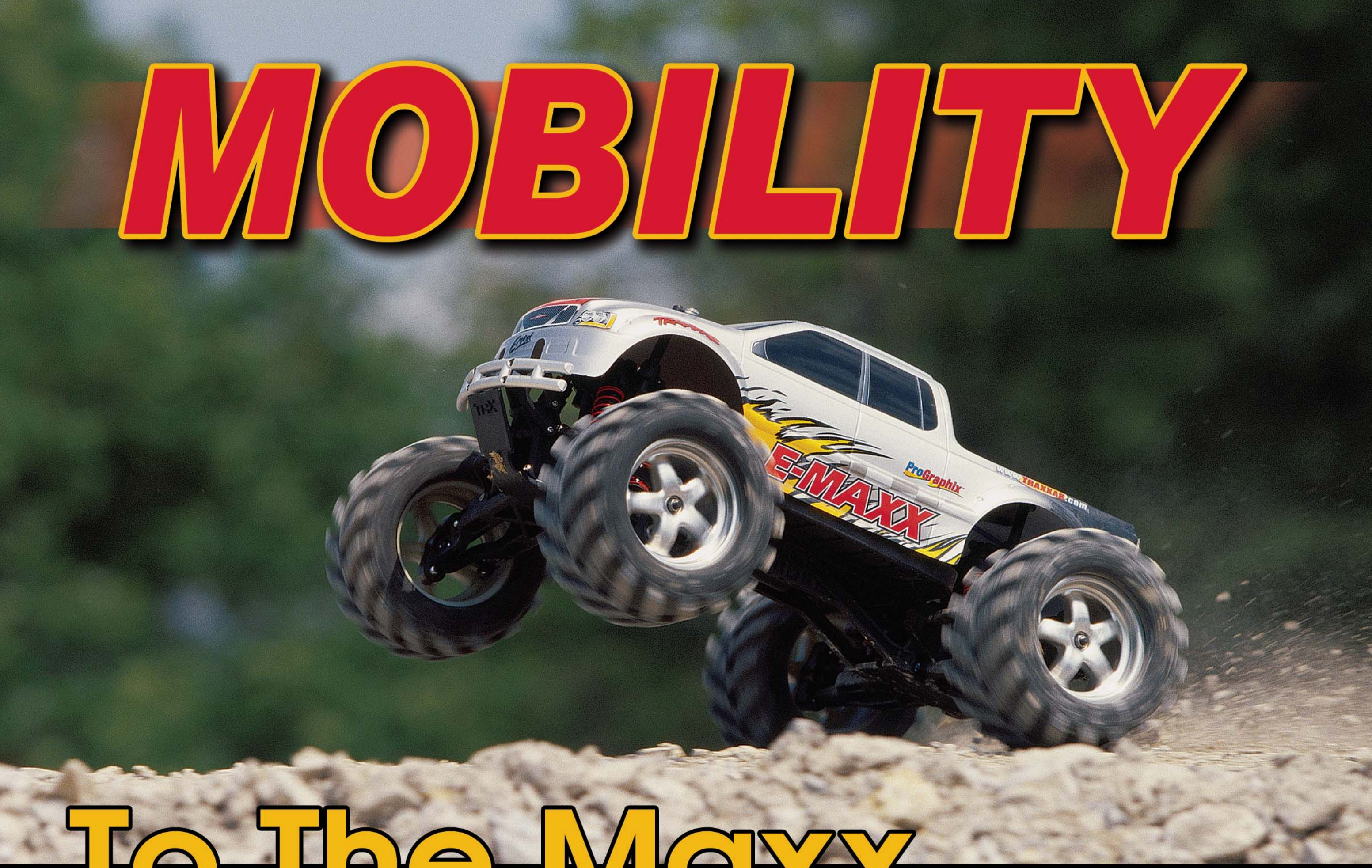
Formula: Pitch Diameter (PD): PD = ND – (.6495 * P) Example: .3125 - (.6495 * .0555) = .2764". Note: This value can be found in the *Machineries Handbook* and is the value meaured by a thread micrometer.

Formula for calculating the diameter of the tap drill (if you don't have a tap drill chart): ND — P Example: .3125 - .0555 = .257 = drill F.

Minor Diameter (MD): MD = ND – (1.0825 * P) Example: .3125 – (1.0825 * .0555) = .252.

Lorne Wilkins

MOBILITY



To The Maxx

PART 4 — A Sense of Direction

by Chris Cooper

For more information on this product, visit **www.machinebus.com/emaxx**

Last month, in our quest to create the coolest and fastest mobile robot on the planet, we gave the E-Maxx improved motor control and some new capabilities. Now it can move at very low speeds and maintain a constant speed in almost any type of terrain. It can also record the distance traveled and move at a specified distance and velocity.

In this article, we'll give the E-Maxx a sense of direction. We'll work through some odometry calculations and add a digital compass. Then we'll demonstrate the navigation method known as dead reckoning — calculating the current position based on the distance and direction traveled from a known position. Finally, we give the modified E-Maxx a trial run and see how well we can navigate around waypoints and return home.

Odometry

Odometry involves using information about the rotation of the wheels to calculate distance traveled. Since the encoder reads the rotation of the spur gear, it cannot measure wheel rotation directly. We must calculate wheel rotation using the overall reduction information — how much the wheel axle turns for each turn of the motor shaft — from the E-Maxx gear chart in Figure 1.

Overall reduction is a combination of the spur and pinion gear ratio, the internal transmission ratio, and the axle ratio. With the encoder mounted onto the center shaft of the spur gear, we need to remove the spur and pinion gear ratio from the overall reduction in order to get the desired transmission ratio. Once we have that ratio and the diameter of the tires, we can infer how

Photo Above: The E-Maxx RC monster truck makes an excellent robotics base. *Photo courtesy of Traxxas.*

far the E-Maxx travels for each count of the encoder. See the sidebar for the calculation details specific to our E-Maxx.

It's important to note that the distance traveled is an approximation. It does not take into account real-world factors such as wheel slippage or gear backlash. The difference between calculated and actual values is small, but can add up over time. One way to compensate for the accumulated error is to attach additional navigation sensors, especially those that use an external reference. A digital compass is not subject to accumulated errors because it is using a fixed external reference — the Earth's magnetic field.

Adding a Digital Compass

For thousands of years, navigators have used the magnetic compass to help find their way. The digital compass we use works on the same principle. The Earth's magnetic field is a dipole, with one magnetic pole near the geographic North Pole and the other near the geographic South Pole. The magnetic poles differ from the geographic poles — which are centered on the Earth's axis of rotation — by about 11.5 degrees (see Figure 2). The magnetic field strength on the Earth varies with location and covers the range from about .1 to 1.0 Gauss. It is this value that we measure to determine the direction of the field. Once we know which direction is magnetic North, we can use that value to determine our current heading.

Figure 1. Traxxas Gear Chart showing overall reduction for different gearings.

Spur Gear Chart

Pinon Teeth		64	66	70	72
12	1st		43.36	45.99	47.30
	2nd		26.92	28.56	29.37
13	1st		40.03	42.45	43.66
	2nd		24.85	26.36	27.11
14	1st	36.04	37.17	39.42	40.55
	2nd	22.38	23.08	24.48	25.18
15	1st	33.64	34.69	36.79	37.84
	2nd	20.89	21.54	22.85	23.50
16	1st	31.54	32.52	34.49	35.48
	2nd	19.58	20.19	21.42	22.03
17	1st	29.68	30.61	32.46	33.39
	2nd	18.43	19.01	20.16	20.73
18	1st	28.03	28.91	30.66	
	2nd	17.41	17.95	19.04	
19	1st	26.56	27.39	29.05	
	2nd	16.49	17.01	18.04	
20	1st	25.23	26.02		
	2nd	15.67	16.15		
21	1st	24.03	24.78		
	2nd	14.92	15.39		
22	1st	22.93	23.65		
	2nd	14.24	14.69		

Do not use with Titan motors.

Overall Reduction

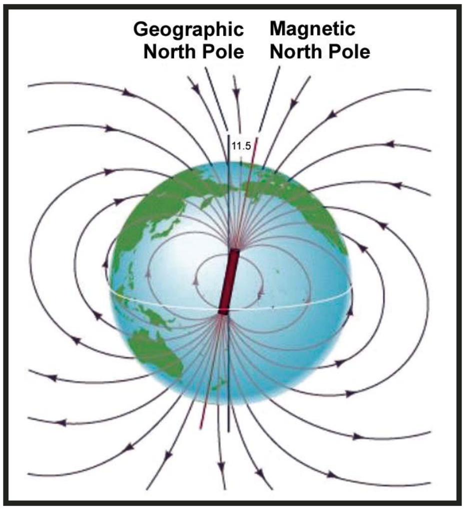


Figure 2. Earth's magnetic and geographic poles differ by 11.5 degrees.

A compass is an inexpensive and effective solution for determining heading, but its readings can be subject to interference. Ferrous metals like steel, nickel, and iron will distort magnetic fields by attracting them. Un-magnetized ferrous materials nearby produce "soft-iron" effects. Magnetized ferrous materials produce "hard-iron" effects.

The E-Maxx itself can have these effects on our compass. For example, the operation of the motors which contain magnets can produce "hard-iron" interference and the NiCad batteries which contain nickel can produce "soft-iron" interference. We can compensate for the "constant" soft and hard interference coming from the E-Maxx through careful calibration. This calibration will not compensate for interference external to the E-Maxx, but if the external interference is temporary, a compass will still be a useful addition.

CALCULATING DISTANCE TRAVELED

The E-Maxx gear chart gives us the overall reduction ratios for first and second gear for each combination of spur and pinion gear sizes. For a 12-tooth pinion gear and a 72-tooth spur gear, the chart gives the overall gear ratio as 47.30 in first gear, which means the motor rotates 47.3 times for each tire rotation.

In order to determine how many times the center shaft of the spur gear rotates, we need to factor out the gear ratio component contributed by the spur and pinion gears. Spur and pinion ratio is calculated using teeth which gives us 72:12, and can be reduced to 6:1. Dividing the overall ratio of 47.30 by the spur/pinion ratio of 6 gives us 7.88 as our transmission reduction in first gear. The encoder disc rotates 7.88 times for each tire rotation.

To determine how far the E-Maxx moves for each tire rotation, we need to calculate the circumference of the tire. I measured a diameter of 5.75 inches, which gives us:

Circumference = Pi * 5.75 inches = 18.055 inches

Therefore, each time the encoder disc rotates 7.88 times, the E-Maxx travels 18.055 inches. For our purposes, the more useful value is for each rotation of the encoder disc, the wheel travels 18.055/7.88 = 2.29 inches. With a 100 count encoder disc, the E-Maxx will travel a distance of .0229 inches per count in first gear. If you do the calculations for second gear, the distance traveled is 0.0369 inches per count. Determining how far you've traveled then becomes:

Total distance traveled = Encoder count * distance per count

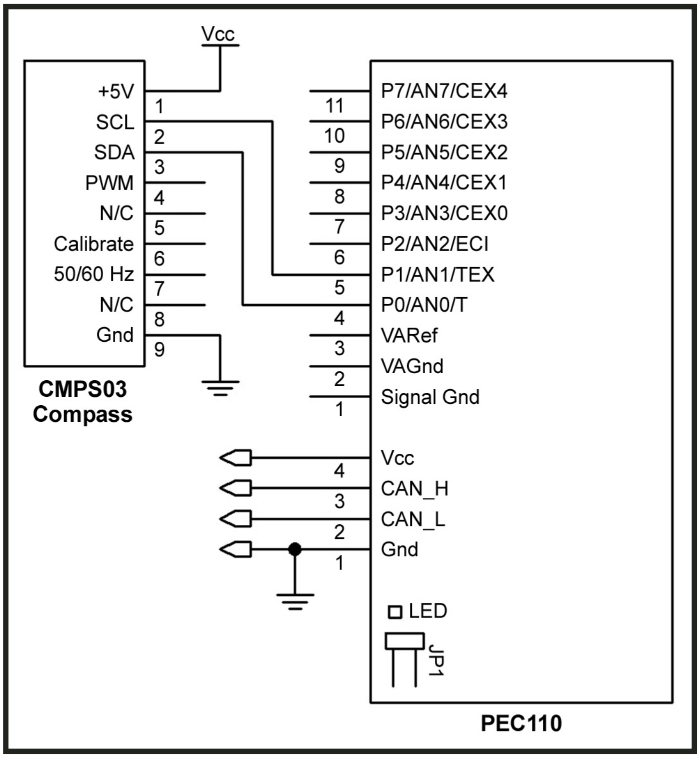


Figure 3. Schematic showing wiring of the CMPS03 Digital Compass to the PEC-110 Port Extender.

Building a Navigation Module

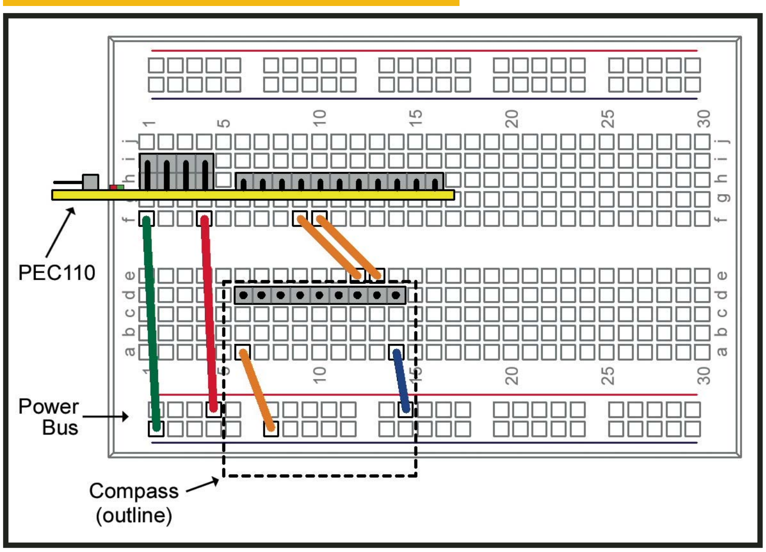
The Devantech CMPS03 digital compass is the first sensor to be added to our newly created navigation module. It is wired to the navigation module, as in Figure 3, and communicates with the new PEC-110, added to the navigation module over an I^2C (Inter-Integrated Circuit) bus. The CMPS03 returns the bearing as a value between 0 and 3599, representing 0-359.9 degrees with 0 degrees being North, 90 degrees being East, 180 degrees being South, and 270 degrees being West.

When installing the compass, I noticed some "soft-iron" interference from the E-Maxx itself. I compensated by following the Devantech CMPS03 calibration process, which can be found in the CMPS03 manual.

The calibration process was very easy. It involves setting the compass into a special calibration mode and slowly rotating the compass 360 degrees. All I had to do was slowly drive the E-Maxx in circles until the calibration was complete. Surprisingly, I found no interference occurred from running the motors, even with the compass positioned at various locations on the deck. As a result, I mounted the compass and navigation module directly on the deck, as opposed to using a tower or mast to isolate it from interference (see Figure 4).

There are only two functional operations: calibrate and getBearing. The "calibrate" method sets the compass into its calibration mode. The "getBearing" method returns the bearing just as the compass returns it, as an integer between 0 and 3599 (see Listing 1).

Figure 4. Diagram of the newly created navigation module with plenty of space left for more sensors.



Dead Reckoning

Dead reckoning involves estimating current position based on the distance and heading traveled from a previously known position. By combining measurements from the encoder and the compass, we can follow a set of waypoints. To test out the dead reckoning capabilities of the E-Maxx, I set up two courses, as shown in Figure 5.

The first course was an oval lap around two cones. From the starting point, the E-Maxx moves 10 feet heading due East (90 degrees), then turns counter-clockwise to a heading of due West (270 degrees) and moves 20 feet, turns another 180 degrees to fast due East again and moves 10 feet to arrive back at the starting point. The second course requires the E-Maxx to navigate around three cones set up in an equilateral triangle.

Steering control is accomplished using a PID (proportional, integral, derivative) algorithm applied to the heading. The PID algorithm works to force the heading error to zero so that the E-Maxx is always pointing in the desired direction and does not overshoot turns.

The course is divided into a series of legs, with each leg consisting of a heading and a distance to travel on that heading. Once the E-Maxx travels the distance,

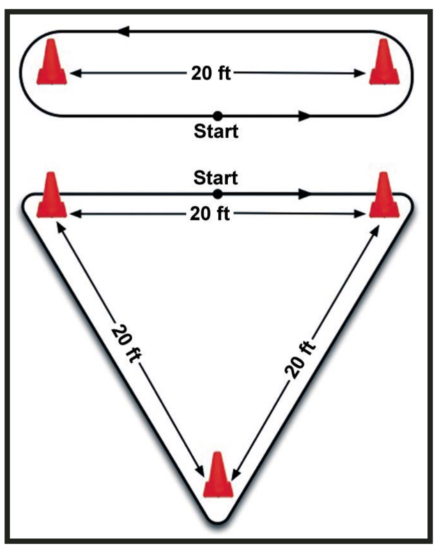


Figure 5. Simple courses test the E-Maxx's navigation ability.

PARTS LIST

- PEC-110 — **www.machinebus.com**
- Devantech Compass CMPS03 — **www.robot-electronics.co.uk**

LISTING 1

```
#include <stdlib.h>
#include <unistd.h>  /* UNIX standard function definitions */
#include "machineBus.h"
#ifndef COMPASS_
#define COMPASS_

    struct compass_t;
    typedef struct compass_t *Compass;

    // Create a new compass reference
    Compass compass_createCompass(CommBus C, uint8_t id);

    // Calibrate the compass
    uint16_t compass_calibrate(Compass C);

    // Get the bearing
    uint16_t compass_getBearing(Compass C);

    // Dispose of the compass
    void compass_disposeEncodedMotor(Compass C);

    uint8_t compass_messageCallback(void* object, CAN_MESSAGE *rxMessage);

#endif /*COMPASS_*/
```

the next leg is retrieved and the E-Maxx slowly turns until it reaches the new heading. When the new heading is reached, the E-Maxx breaks out of the turn and begins traveling the distance of the next leg. This allows me to simplify calculations and ignore the distance traveled in the turn, yet still accurately traverse the course. The code in Listing 2 demonstrates this approach.

After running the first course, the E-Maxx was off by just three inches. The longer distance traveled and the additional turns in the second course caused the accumulated error to increase to seven inches.

Conclusion

Navigation is a difficult problem, and I've barely scratched the surface. But our navigation module is off to a good start. We have shown that dead reckoning using the encoder and the digital compass is an effective way to navigate over short distances. As distance traveled and number of turns increased, so did our positional error. Nonetheless, the results indicate just how accurate dead reckoning can be.

If dead reckoning was to be our only method of navigation, we would want to be more precise with respect to turning by taking into account turning radius, distance traveled in the turn, and the velocity of the E-Maxx through the turn. However, we have different navigation techniques in store.

In the next article, we will be demonstrating GPS navigation. GPS navigation is an alternative, but complimentary navigation technique that is useful when you know the longitude and latitude of where you need to go. GPS readings are not subject to cumulative errors, so it's a great way to accurately navigate over longer distances. We'll tackle GPS navigation by installing a GPS receiver and following a trail of GPS breadcrumbs just like the competitors in the DARPA Grand Challenge. **SV**

LISTING 2

```
leg = courseIterator->getNext();
while ( !done && leg != null) {

    /* Check Bus status */
    if (commbus_status(bus) != 0) {
        printf( "Failed to retreive status\n" );
    }
    if ( commbus_readyToTransmit(bus)) {

        // Get our current bearing
        currentBearing = compass_getBearing(compass);
        // Pass in leg heading and current bearing into
        // steering controller to turn to correct heading
        steeringcontroller_steer(steering, leg->getHeading(), currentBearing);

        // if we are way off (more than 10 degrees) then assume we are at a turn
        // start doing a slow turn
        if (abs(currentBearing - leg->getHeading()) > 100 ) {
            encodedmotor_setRate(motor, 150);
        } else if (newLeg) {
            // Start of new leg, heading close so set the distance to travel
           encodedmotor_resetCount(motor);
           encodedmotor_setAbsolutePosition(motor,
                                            500, countFromDistance(leg->getDistance() );
           newLeg = 0;
       }
    }
    // If we've traveled the distance, get the next leg
    if (encodedMotor_getCount(motor) >= countFromDistance(leg->getDistance()) {
        leg = courseIterator->getNext();
        newLeg = 1;
    }

} // while
```

RESOURCES

- Autonomous E-Maxx: **www.machinebus.com/emaxx**
- Rossum Project Papers — An excellent source of information on robot navigation: **http://rossum.sourceforge.net/papers**
- Philips KMZ51 Application Notes: **www.semiconductors.philips.com/pip/KMZ51.html**

by Simone Davalos

Power Tool Drag Racing

Sunday! Sunday! Sunday!

Do you know what I like about competitions? Certainly camaraderie is at the top of the list. Innovation, skill, and cooperation fall in there somewhere too, I'm sure. Of course, there's also the thrill of two teams competing and only one winner. However, there are often a few unsung advantages that really get taken for granted at most events. People rarely consider how great it is to have things like shiny new auditoriums. Or immaculate food service. Or, you know, running water and electricity. One very rarely goes to an event with the fear of tetanus, ptomaine poisoning, or being accidentally sprayed in the face with a shotgun.

This is why we have The Power Tool Drag Races.

What is Power Tool Drag Racing, you ask? Why, exactly what it sounds like, my friends. Teams compete for honor and glory by racing different classes of machines down a 75-foot long track, in a one-on-one no-turns-required speed race. Machines range from straight-out-of-the-box super stock power tools all the way to Unusual Design/Top Fuel mutated monstrosities that may or may not have had some part of them involved in something that could be loosely defined as a power tool at some point in the past (did you get all that?).

All of this takes place amidst cheering crowds of sunburned, drunken, nerdy gearheads. The location is a real, working junkyard, specially tidied up for the event. Never fear, however, there are still plenty of opportunities for severe injury and nasty infection. This is why all audience members are made to sign a waiver before entry.

At the Ace International Speedway (also known as Ace Auto and Scrap) on a beautiful day in San Francisco, CA this past May, hundreds of Power Tool Race enthusiasts gathered for another installment of what one announcer called "The Death March of Fun." Fifty-nine racers were competing in four classes to see who went home with the glory and the prize money, and who remained to drown their sorrows in beer and 3-in-1 oil. Ages ranged from six weeks to 85 years old, making it truly fun for the whole family.

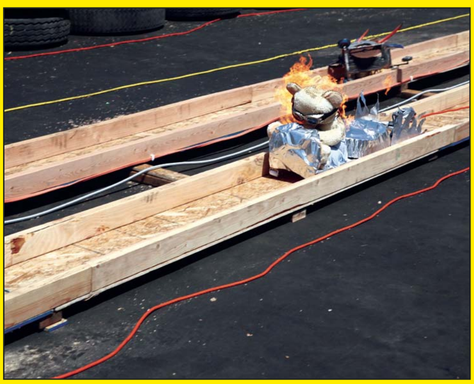
Power Tool Drag Racing has been an institution in San Francisco since its inception five or so years ago. It's been covered as a four-episode series on The Discovery Channel, and has spawned numerous offshoot events in places like Seattle, New Orleans and The United Kingdom. It mostly happens every year, except when it doesn't, as Charlie Gadeken, one of the co-organizers, has been heard to state. Everyone loves power tool racing for the camaraderie and can-do punk rock attitude of the competitors. They also love it for the family-friendly, bloodthirsty, take-no-prisoners competition.

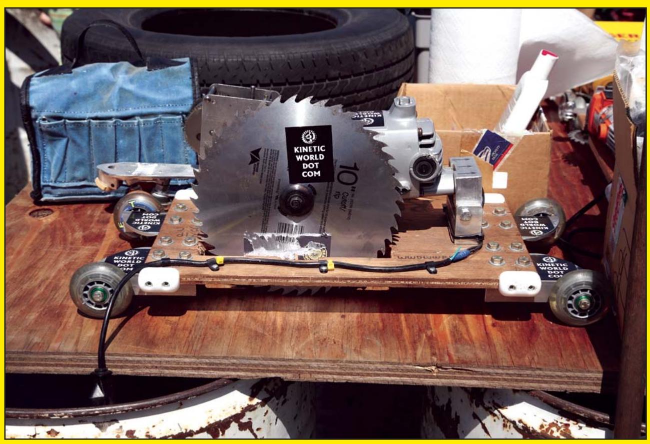
At the crack of noon, the first racers were up on deck as DJ Big Daddy played the traditional Jimi Hendrix rendition of the national anthem. Hats were barely off of heads and many hearts were still covered when the first set of racers went screaming down the track. Thus began a solid 10 hours of racing mayhem.

As the audience entered the event, the Flaming Lotus Girls Admission Auxiliary smilingly separated race-goers from their money in exchange for entry tickets, Official Power Tool Drag Race t-shirts, and racing fees. When Flaming Lotii aren't granting you entry or racing power tools, they spend their time building large-scale fire sculptures for use in places like Burning Man in Nevada and RoboDock in Amsterdam (see more at **www.flaminglotus.com**). You don't have to be female to be a Flaming Lotus Girl, and they are very, very good at what they do. Which is consuming huge quantities of propane and sending jets of flame hundreds of feet into the air (no really — their stuff makes military flame throwers look like prayer candles.)

The classes for the Power Tool Drag Races are divided up much like regular drag racing. Super Stock vehicles are often directly off the shelf from any hardware store. As was discovered by the New England Belt Sander Racing Association (from whom all things power tool racing flow), often the most effective power

> “What is Power Tool Drag Racing, you ask? … Teams compete for honor and glory by racing different classes of machines down a 75-foot long track, in a one-on-one no-turns-required speed race.”





tool racer is a plain and unadorned belt sander. Sometimes super stock machines can be outfitted with custom gear, chain or tire configurations, but as long as the motor and the power supply came out of a plain old power tool of some sort, it's acceptable.

The next class is the Pro-Superchargers. These are the same as the super stock, but with more than one motor, and don't think the contestants don't make the most of it!

The Awful Awful Altereds class has the most room for gearhead artistic improvisation. All motor modifications and power source fiddling is legal, as long as it definitely is recognizable as having parts that come off a power tool.

The category that has caused most arguments among competitors is the Unusual Design/Top Fuel class. In years past, there have always been a few contenders that didn't quite fit the standards of the class in which they were entered. Maybe their power sources were incredibly unorthodox (fire extinguishers, gas canisters, 12-gauge shotguns); maybe they were machines that only very loosely fit the interpretation of "power tool" (a sling shot). At any rate, to avoid any more knock down, drag out arguments among officials and competitors (though the organizers do always like a halftime show), a new class was created specifically for this sort of machine. These machines tend to be the most dangerous, and in a world where no one blinks as a fire-spitting tricycle in the Awful Awful Altered class launches itself off the track and into the audience, some of them are dang scary. This is the only class where officials reserve the right to pull the machine from competition without warning if it seems like it will be too damaging to the audience and the organizers' insurance premiums.

The Funny Car class is for custom fabricated machines that carry one or more rider(s) down the track, using one or more motors that originated in a hand tool. You may think this might limit your ability to go as fast as possible, but it just depends on how innovative you get in finding your hand tools. Team Inertia Labs, long-time PTDR competitors, has used an industrial slaughterhouse saw for one of their funny cars, nicknamed "The Heifer Halfer."

The crowd remained excited and well lubricated as the event wore on throughout the day. The Official Race Flag Girl kept time for the announcers, while a digital sign related race times to the judges. Only a few audience members at the end of the track had to dive out of the way of a hurtling machine or two, and the bartenders were up to their necks slinging beer and Junkyard Hot Dogs (cooked with a steam iron on an aluminum foil tray! Tasty!).

Team Korntee's first run with the super stock racer Buffer the Vampire Slayer (a specially modified shoe-buffer with wheels) ended badly when his drive buffer couldn't get enough purchase on the track and spun itself apart. Buffer was pressed into service later on at the bar to keep everyone's combat boots bright and shiny.

Longtime old hands at Power Tool Racing, father and son Lowell and Steven Nelson, arrived with several of their machines in fine form. After a few fraught moments and a few stunning races, they went home with first and second place in the Super Stock Category, an outcome that surprised precisely nobody. Their racers, DeWalt Assault and Magdelana zipped down the track in a hotly debated final run, which awarded Lowell with First Place and Steven with Second. What a way to spend a 45th wedding anniversary!

Huh? Wedding anniversary? Yes, Lowell had brought his lovely wife to PTDR, where she spent the last four hours of the event sitting in the passenger seat of the car, waiting to go home. Lowell got an award for that one ... (see sidebar).

The Funny Car competition was chock-full of innovative racing technology. Along with old favorites like Team Phoenix (The Phoenix Chainsaw Chopper) and Team Washburn (Drag Queen), there were a few notable newcomers. Team Saw Werks brought forth a stunning display of warranty-voiding machinery named Stihlborn. Stihlborn consisted of a kiddie bicycle with a Stihl chainsaw motor with a custom-built nitrous oxide and gasoline injection system. Good for racing and doing donuts in the 7-11 parking lot!

One of the really innovative ride-ons was Team Inertia Labs, with their all-pneumatically powered racer Thor. Thor is built with a pair of nitrogen tanks compressed to 2,500 psi and two vintage pneumatically-powered skil-saws. The saws were designed to run

PTDR 2006 CHAMPIONS

Super Stock
1st — Dewalt Assault, Dash4Cash
2nd — Magdelana, Team KISS

Pro-Supercharged
1st — Double Barreled, Team CTP
2nd — Monorail! MonoRail!, Team Impotence

Awful, Awful Altereds
1st — NeoCon Outlook, Team Schneeveis
2nd — Monorail! MonoRail! Monorail!, Team Impotence

Funny Car (Ridden)
1st — Thor, Inertia Labs
2nd — Matt Dawg Express, Team Plumb Crazy

Unusual Designs/Top Fuel
1st — Big Bang Barbie, Team Schneeveis
2nd — fox1, Team Fox

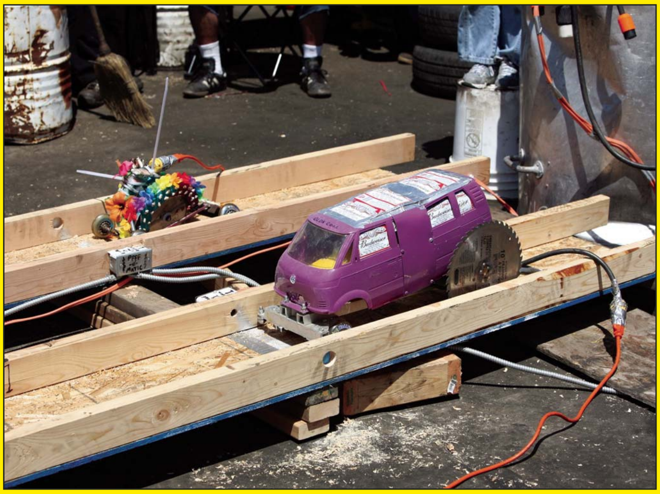
Sex Toys
1st — Scratch, Thingum
2nd — LPD, Team LPD

Worst Engineering
Paul the Plumber

Cutest Contestant
Charlotte Egeria

Builder Least Likely to Get Laid
Lowell Nelson

***Safety Third* Victim Award**
Jonathon Foote



Tim the Toolman has nothin' on this crowd. A good — and reasonably safe — time was had by all. All photos are courtesy of Scott Beale.

at 250 psi, which made forcing 1,000 psi through them to race super exciting. Added to this, the brakes failed on the last run of the day, nearly making Shane (aka Bird) winner of the PTDR "Safety Third" Award. Sadly, he didn't place in the Safety Third category.

Team Schmokin' represented for the Pro-superchargers with style points, if nothing else. Their racer, Schmokin' #1 consisted of four drill motors with saw blade wheels, mounted on the bottom of a working propane grill. Enough cup holders for two cans of Tecate completed the machine as it sped down the track, making hot dogs as it raced. Mmm, meat by-products.

The Flaming Lotus Girls incinerated the competition with Bunny Shark Mini, two four inch angle grinders mounted on a shocking pink frame, shiny wire-wheels for decoration, and the pre-requisite flame thrower on top. Girl power never looked so good.

All was not always fun and games, however. Doctor Jonathan Foote (a real PhD!) of Rotorbrain Industries (**rotorbrain.com**) was the winner of the annual PTDR "Safety Third" Award for Excellence In Not Getting Killed, when an Unusual Design machine ran slightly amok, resulting in a teensy bit of buckshot to the face. Not enough to make Dick Cheney jealous, but enough to give Nosferatu the munchies. Judge Dave, PTDR Statistician and All Around Official, comments "I had just enough time to say 'Hey! That thing looks like it's powered by a 12 gauge shotgun!' when the thing went off." Dr. Foote sustained minimal damage and emerged from the EMT's care with high spirits, commenting "It's okay, chicks dig scars."

All in all it was a successful event, with just enough catastrophe to make the rest of the day interesting.

Just like NASCAR ... or not.

Power Tool Drag Racing In 2006/2007

Come on guys, this is not rocket science. All you need is a power tool and the need for speed. As Steven Nelson, longtime power tool racing champion says, "I can build a power tool racer in under a day. Can you?"

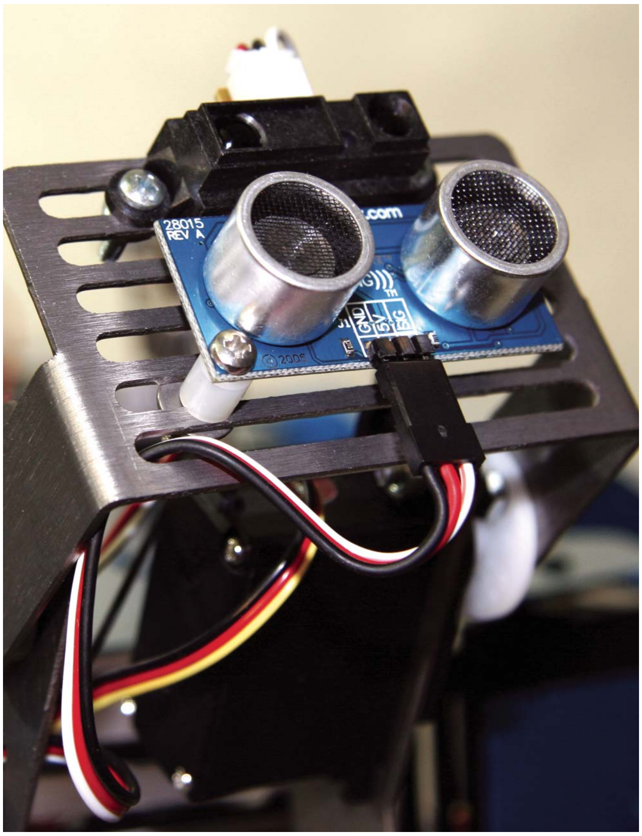
Full rules for Power Tool Drag Racing are available on the site **www.powertooldragracing.com** Guidelines for starting a PTDR in your area are on the site as well, why not take a crack at it yourself?

The event in San Francisco is usually in May or June every year, however the date does tend to shift a bit. Plus or minus six months. Check the website for details. **SV**

AUTONOMOUS ROBOTS

and Multiple Sensors

Part 1: FUSION FUNDAMENTALS



We derive a sense of self and of our environment from thousands of sense organs, ranging from our eyes and ears to the golgi tendon organs that monitor our muscle contractions. Similarly, autonomous robots employ multiple sensors of different types to assess their environment and internal state. Proprioceptive sensors range from pressure sensors under the feet of hexapods and potentiometers attached to the joints of robot arms to the wheel encoders used on carpet rovers. Environmental sensors include the ubiquitous ultrasound and infrared rangefinders, as well as the more exotic humidity sensors, accelerometers, gyroscopes, and GPS receivers.

Although it's possible to go overboard with the number and type of sensors, in practice, cost, weight, space on the robot chassis, and availability of onboard or network bandwidth and processing power limit the sensor population. Even so, thanks to more powerful and affordable sensors, multi-core microcontrollers, and remote PC processing, the robotics community is moving *en masse* to multiple sensor autonomous robot configurations. This first article of two explains how multiple sensors can be used to enhance the performance of autonomous robots and introduces the concept of sensor fusion.

PHOTO ABOVE. The infrared GP2D12 and ultrasonic Ping))) rangefinders mounted on the tilt-pan head of a hexapod.

NOTE

The code listings mentioned in this article are available on the *SERVO* website at **www.servomagazine.com**

SENSOR BASICS

A robot bristling with sensors doesn't necessarily perform any better than an inexpensive carpet rover equipped with a single IR sensor. As Braitenberg notes in *Vehicles*, even a robotic Cyclops, when properly configured, exhibits lifelike behavior [1]. Conversely, although it may look impressive, a robot with an abundance of sensors only guarantees expense, significant battery drain, and computational overhead.

Multiple sensors, properly configured, can enhance autonomy if they reflect a robot's mission, physical structure, and operating environment. Just as the environment selects for the fittest organisms in nature, the algorithms used to control sensors and process sensor data determine how well a robot will perform in a given environment.

Limitations

The field of robotics has been transformed by the availability of affordable, powerful, and intelligent sensors. Instead of working primarily with raw transducers, roboticists have angular rate gyroscopes, GPS receivers, accelerometers, and solid-state compasses at their disposal. Even entry-level robot kits feature compact IR detectors and microcontrollers.

However, every sensor has limita-

by Bryan Bergeron

tions due to design and manufacture and from interaction with the environment. Sensors fail and change with time due to thermal settling, long-term aging, and physical or electrical damage.

Whether a sensor exhibits drift or blatant failure, the result is the same. The host robot is hampered in its ability to function in the environment — that is, unless provision is made for faulty sensors. Similarly, the variables in the environment and within the robot that are indirectly quantified by sensors are not completely knowable.

As such, there is always a stochastic (random) component of distance, position, temperature, or other parameter measured by a sensor. This stochastic nature of sensor data is magnified when coupled with limited sensor accuracy and fidelity.

Variability

Sensors — like most other electronic components — are manufactured to certain specifications, to suit a variety of applications and markets. Even the best sensors are produced with some degree of variance from the ideal. A common means of minimizing uncertainty in data due to inherent sensor variability entails developing a meticulous model of the sensor characteristics relative to the known environmental variations. The goal is to characterize sensor response in likely environmental situations, such as ambient temperature, humidity, light level, proximity of metal, or magnetic fields, as well as the passage of time.

To illustrate variability in sensor data caused by the environment, consider the popular ultrasound sonar sensor, typified by the Ping))) sensor from Parallax. Like the Devantech SRF-04 and several other self-contained ultrasonic rangefinders, the Ping))) sensor operates by emitting a 40 kHz pulse and timing the return echo [2]. The sensor produces a TTL-level output pulse that has a width corresponding to time required for the pulse to travel to the target and back again.

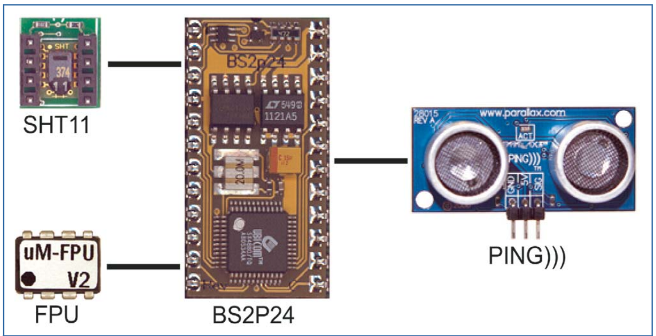


FIGURE 1. Configuration of electronic rangefinder components.

Armed with calibration data, it's possible to correct the sensor's time readings to suit a particular environment. For example, temperature, and, to a lesser extent, relative humidity, affect the accuracy of distance calculations based on data from the Ping))) sensor.

Temperature and humidity should be considered when working with ultrasonic rangefinders because they affect the speed at which the 40 kHz pulse travels from the ceramic transducer element to the object and back to the receiver element of the sensor. The nominal speed of sound in air — 1,130 feet per second — is an approximation for dry air at room temperature, at sea level, and with a typical CO_2 concentration. A more accurate figure considers the temperature, pressure, humidity, and CO_2 concentration, in the form of the following equation:

$$V_{sound} = (nRT/M)^{1/2}$$

where n is the adiabatic constant, characteristics of the gasses in air (nominally 1.4), R is the universal gas constant (8.314 J/mol K), T is the absolute temperature in Kelvins, and M is the molecular weight of the gases in kg/mol. In other words, the speed of sound in air is proportional to the square root of the absolute temperature, and increases slightly with increasing humidity and CO_2 concentration [3].

Although technically correct, solving the equation with reasonable speed and accuracy is problematic using integer arithmetic on a typical microcontroller. A more microcontroller-friendly model for the velocity of sound in air is:

$$V_{sound} = 331.4 + 0.6\ T_C \text{ m/s}$$

where T_C is the temperature in degrees Celsius. The 331.4 figure is the speed of sound at 0 degrees Celsius. The small contributions of humidity, air pressure, and CO_2 content are ignored in this model.

A circuit configuration that compensates for the variability in ambient temperature during the operation of an ultrasound electronic rangefinder is shown in Figure 1. A Sensirion SHT11 temperature and humidity chip and Micromega uM-FPU floating point coprocessor — both available from Parallax — are used to provide real-time temperature compensation.

Listing 1 is the PBASIC program of an electronic rangefinder based on a Parallax BS2p24 and Ping))) sensor. The components are connected as per the component sheets available on the Parallax site (**www.parallax.com**), using the I/O assignments provided in Listing 1.

Following constant and variable declarations and initialization of the temperature sensor and floating point unit, the ambient temperature is read from the SHT11. Temperature data is then applied to the velocity approximation formula given above, and the results are displayed in the PBASIC development environment using the DEBUG function. Both uncompensated and compensated distances are computed and displayed.

Tests in a room held at 21 C revealed that the compensated figure at times varied from 1 to 1.5 cm from the uncompensated distance measure. More importantly, both measures

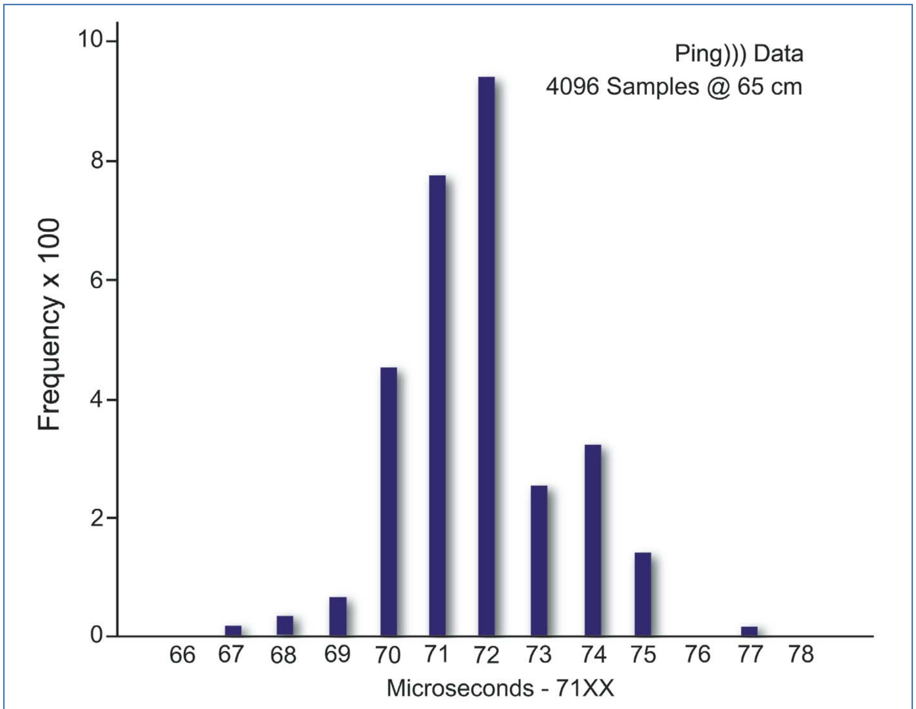


FIGURE 2. Frequency distribution of raw Ping)))™ time data output.

showed variability from one measurement cycle to the next when repeatedly measuring the distance between the Ping))) sensor and a wall. Figure 2 shows the uncompensated time data produced by the Ping))) over 4096 consecutive measures. The range of measures — from 7166 to 7178 or 16 microseconds — corresponds to a distance range of about 0.25 cm.

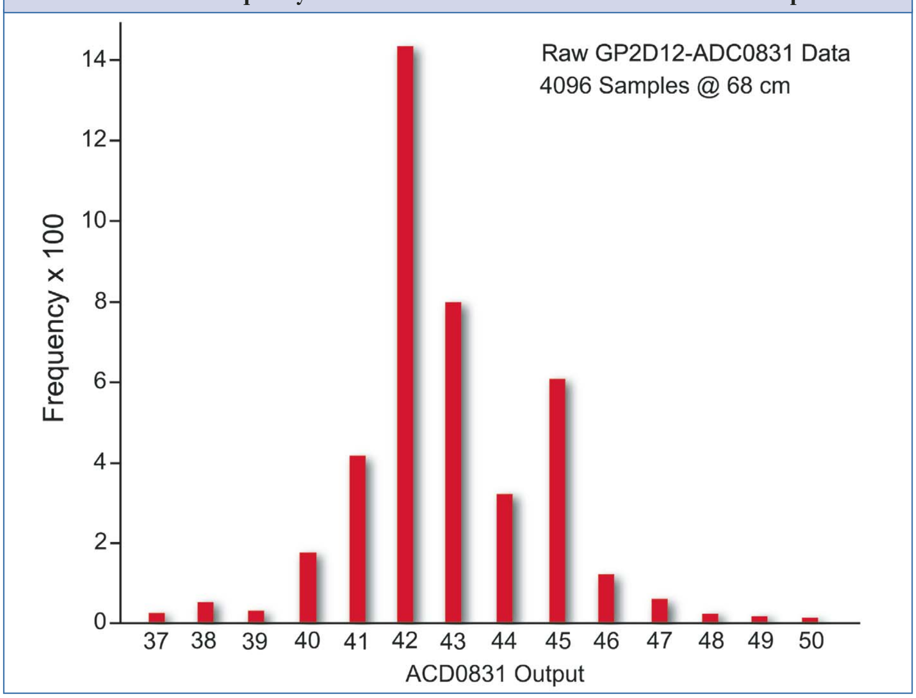
According to the specifications from Parallax, the Ping))) is capable of detecting target distances from about 3 m up to 300 cm, with a maximum polling spacing of 65 µS or about 15 kHz. The Micromega uM-FPU floating point coprocessor is used to compute the adjusted distance, and to illustrate another approach to maximizing the accuracy of measurements.

Although the BASIC Stamp provides several workarounds for integer math, such as the ** and */ operators, most other microcontrollers provide support for floating point arithmetic. There is another reason for introducing the uM-FPU in the hardware design, which will be discussed shortly. A robot equipped with this temperature compensation feature should perform equally well measuring distances inside or outdoors, within the operating temperature and humidity range of the Ping))).

A halogen desk lamp and can of compressed air are useful in testing temperature compensation responsiveness. A few seconds of air can drop the temperature of the sensing chip 10 or more degrees Celsius, and positioning a halogen desk lamp over the SHT11 chip has the opposite effect. Note that during testing, cooling the SHT11 affects the sensor, but not the air between the Ping))) and reflecting object.

Readers interested in further refining the compensation by taking humidity into account should review the paper by Olson Cramer [3] and the code posted by Tracy Allen of EME systems [4]. Conversely, for those interested in a temperature-only sensor, the DS1620 Digital Thermometer is an inexpensive, well-documented solution.

FIGURE 3. Frequency distribution of unfiltered GP2D12-ADC0831 output.



Target Specificity

Another characteristic of sensors is illustrated by considering another commonly used sensor in robotics work: the Sharp GP2D12 infrared rangefinder. Like the Ping))), the GP2D12 emits a signal that bounces off of the target and returns to the receiving element of the device. However, the similarities end there. The GP2D12 is an analog device that relies on triangulation of an infrared beam to measure distance from about 10 to 80 cm.

Furthermore, instead of returning the time interval digitally encoded in microseconds, the GP2D12 produces an analog voltage that is a non-linear function of the distance between the IR emitter and the target. A higher

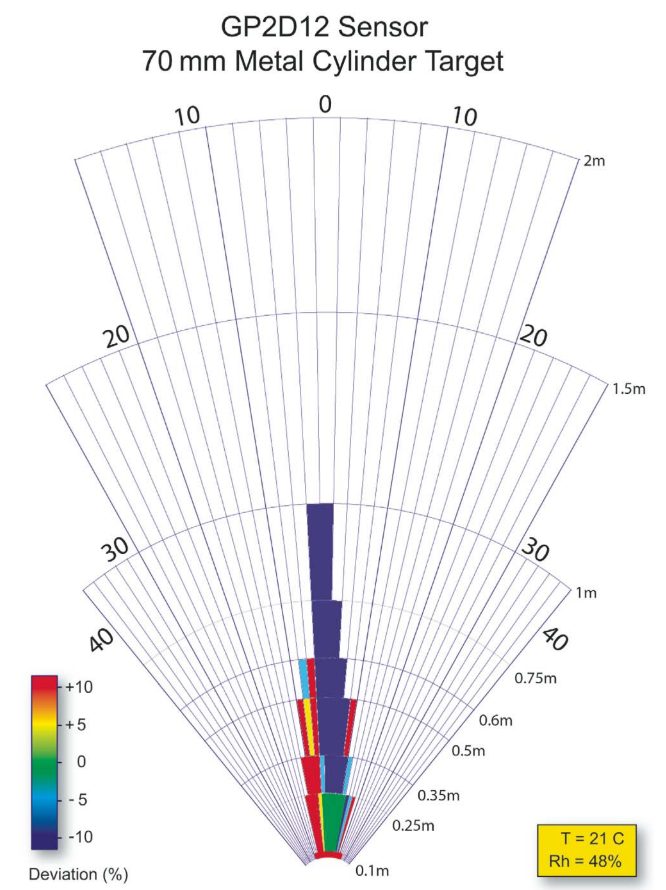


FIGURE 4. Ping))) sensor range and bearing sensitivity for a 70 mm metal cylinder target.

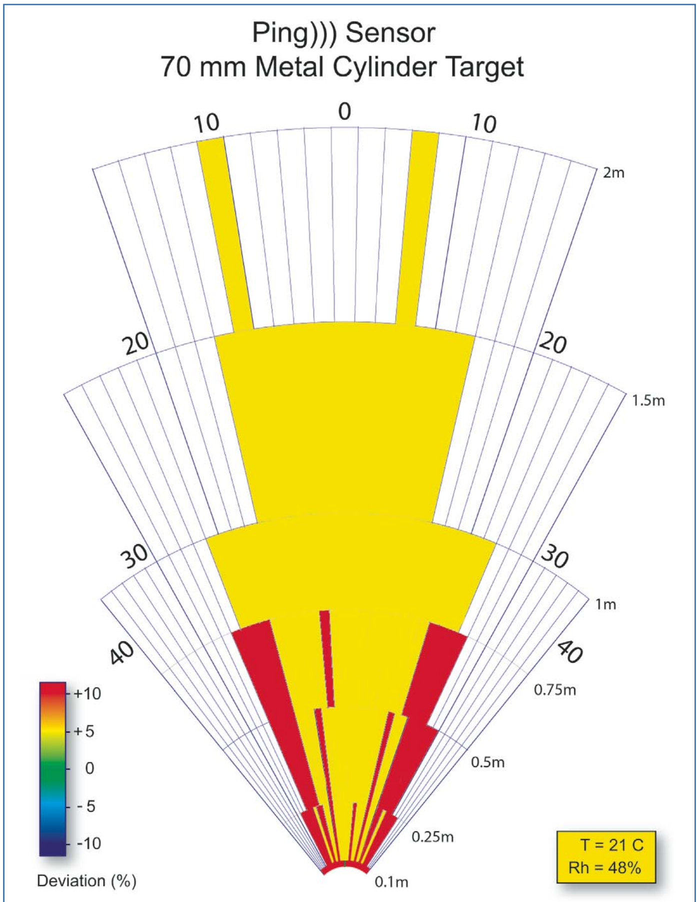


FIGURE 5. GP2D12 sensor range and bearing sensitivity for a 70 mm metal cylinder target.

output voltage corresponds to a smaller emitter-target separation.

The maximum polling frequency of the GP2D12 is significantly greater than that of the Ping))) at 25 kHz, which corresponds to a minimum measurement interval of about 40 µS. The digital version of the Sharp sensor is free of the overhead of the A-to-D converter, but the minimum measurement interval is nearly double that of the analog device.

Listing 2 shows a minimalist program in PBASIC to read a GP2D12 by polling an ADC0831 eight-bit A-to-D converter. The schematic of the standard configuration is available on the Parallax site, under information for the GP2D12, as well as volume 5 of The *Nuts & Volts of BASIC Stamps* [5]. Aside from the BS2p24, GP2D12, and ADC0831, the only other component needed is a potentiometer to provide a 2.5 volt reference for the A-to-D converter.

As with the Ping))), the GP2D12-ADC0831 combination exhibits measure-to-measure variability. Figure 3 shows the frequency distribution of 4096 consecutive samples with a wall as a target at 68 cm. The variability roughly corresponds to 0.5 cm at the distance, or roughly twice the variability of the Ping))). Because of this jitter, the output of the ADC0831 is typically filtered by averaging several consecutive measures [5].

Distance accuracy wasn't considered in this example, but there are at least two sources of error not addressed by the code in Listing 2. The first is variability from sensor to sensor. As an analog device, the nonlinear voltage output curve varies from one sensor to the next. This variability may be insignificant for some applications, but critical in others, depending on the accuracy requirements.

The second source of error is the use of an approximation of a linearization function, in the form of a lookup table, to transform the output voltage level to a distance measure. Although solving a third or fourth order polynomial in real time is beyond the capabilities of a naked STAMP, it's within the limits of one equipped with a mathematical coprocessor, such as the Micromega uM-FPU. Moreover, Micromega offers a floating point calibration program on their website that can be used to create a linearizing function specific to a particular GP2D12 [6].

Given the differences in operating frequencies, construction, and operating parameters, it isn't surprising that the Ping))) and GP2D12 sensors provide different results, depending on the target and the environment. For example, consider a robot world that consists of an expansive garage in which metal cylinders (empty gallon paint cans) and rubber balls are randomly distributed. How will the ultrasound and IR range sensors perform?

To answer this question, a Ping))) and GP2D12 were mounted on a microprocessor-controlled tilt-pan head at a

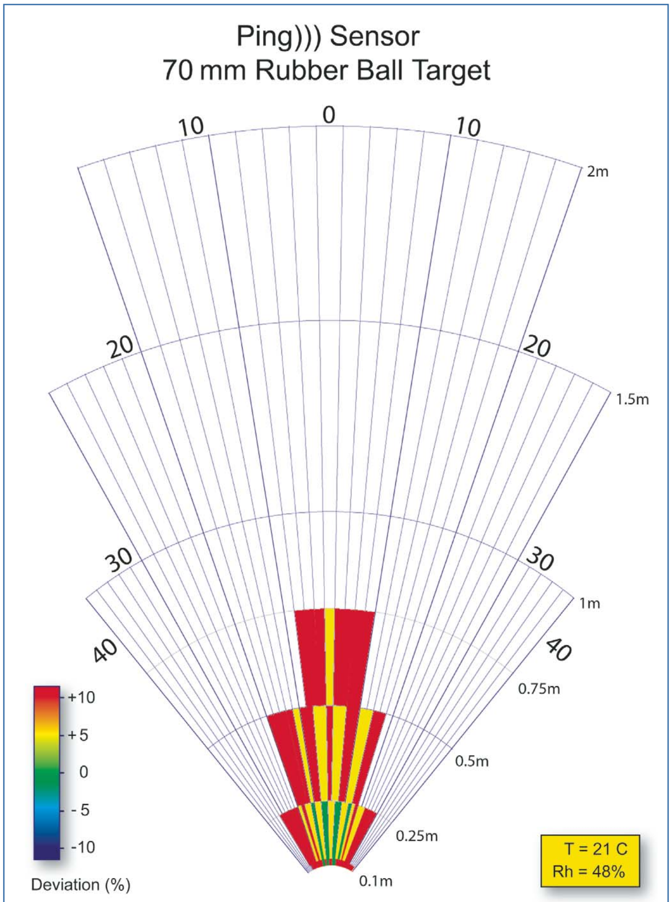


FIGURE 6. Ping))) sensor range and bearing sensitivity for a 70 mm rubber ball target.

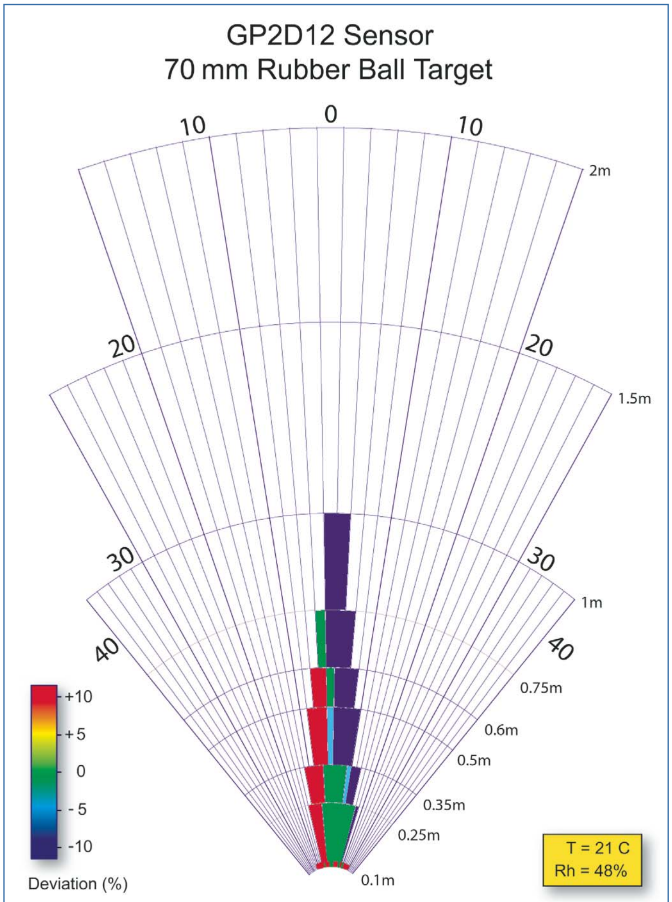


FIGURE 7. GP2D12 sensor range and bearing sensitivity for a 70 mm rubber ball target.

height of one meter. A target — a standard 70 mm diameter empty paint can — was hung at specific distances from the sensor by securing the target to the ceiling at fixed intervals by a monofilament thread. Distance measurements were taken from the sensors at two degree increments at distances of 10, 25, 50, 75, 100, 150, 200, 250, and 300 cm for the Ping))) and 10, 25, 35, 50, 75, and 100 cm for the GP2D12.

Target was centered at one meter, in the same horizontal plane as the sensor, and the ambient temperature and relative humidity were 21C and 48%, respectively. The target was suspended at sensor height to negate the effect of the floor on sensor response and to make the results more generalizeable.

Figures 4 and 5 show the results of the study. The distance data from the Ping))) shows the sensor consistently overestimated the sensor-target distance with the metal paint can, and the maximum range was only 200 cm, and this was side-lobe pickup. Maximum heads-on range was 150 cm, or 50% of the stated range of the sensor. The maximum bearing varied from nearly ±40 degrees with the nearest edge of the can 10 cm from the sensor to approximately ±10 degrees with the target at 150 cm.

The study was repeated with the Ping))) mounted vertically. There was no significant difference, other than a maximum range of 150 cm and no side lobes. In comparison, the GP2D12 — which was programmed with the averaging filter as per Parallax documentation — had a maximum range of 100 cm with the empty paint can target. Maximum bearing varied from about ±25 degrees at 10 cm to 4 degrees at 100 cm. There was no difference with the sensor mounted vertically.

A second study was conducted with a 70 mm Togu-Prien rubber ball as a target. The results — shown in Figures 6 and 7 — are markedly different from the study based on the metal can. The range of the Ping))) was only 75 cm, compared with 100 cm for the GP2D12. Data from the IR rangefinder was roughly equivalent to that from the previous study, with a range accuracy from under to over distance estimation.

Apparently, the sonar signature of the rubber ball was considerably less than that of the paint can. In contrast, the infrared signature for both targets was similar. This is borne out by Figure 8, which shows the high reflectivity of the rubber ball under IR illumination. The compressibility of the rubber ball likely resulted in a smaller signature at the 40 kHz operating frequency of the Ping))) sensor.

SENSOR FUSION

The differences in response of the IR and US sensors, summarized graphically in Figure 9, can be used to our advantage through a process referred

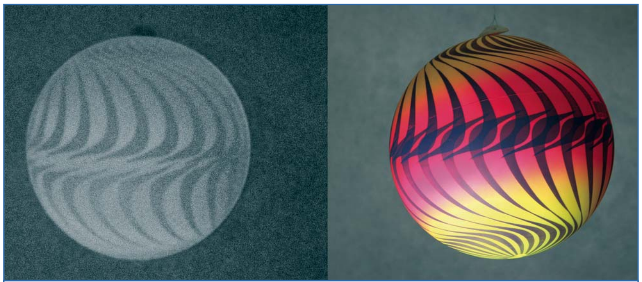


FIGURE 8. 70 mm diameter rubber ball target under IR (left) and visible light (right) illumination.

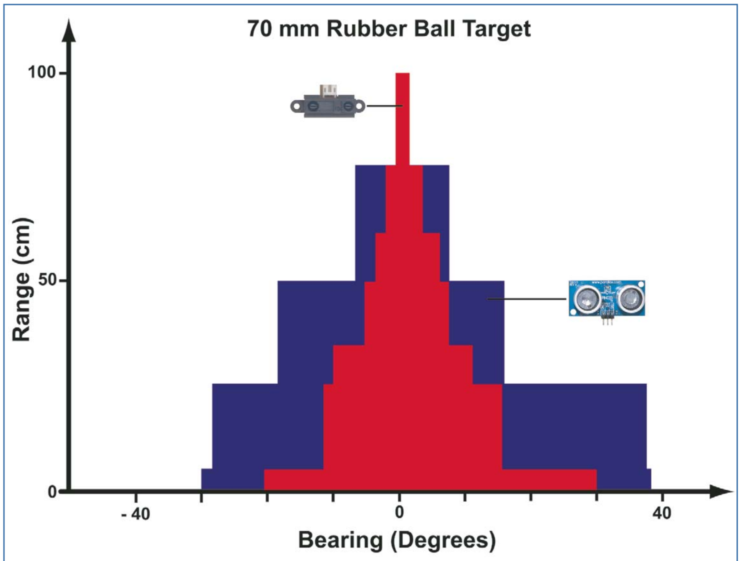


FIGURE 9. Overlap of bearing and range coverage of the Ping))) and DP2D12 with a rubber ball target.

to as *sensor fusion* — the use of data from multiple sensors to decrease the uncertainty of measurement. Sensor fusion can be implemented at the signal, data, feature, or decision level, using either identical or different sensors. Signal-level fusion provides a signal in the same form as the original, but with, for example, greater accuracy and less drift. The signals from two ultrasonic range sensors — both in the form of propagation time — can be fused at the signal-level, for example.

Sensor fusion at the data level involves manipulation of data once it has been normalized to the same form and format. For example, even though the data produced by the Ping))) and GP2D12 are markedly different, when converted to centimeters, the data can be compared and manipulated. Figure 9 shows the overlap of bearing and range for the US and IR sensors, using computed distance data.

In statistical terms, the GP2D12 displays less dispersion, compared with the Ping))). Fusion at the feature level involves specific features of the data, such as range only. Decision fusion operates at an even higher level, and is concerned with, for example, what action to take, based on sensor data.

A simple way to employ sensor fusion at the signal level, whether from sensors of the same type or of different types, is illustrated in Figure 10. Using simply the absence or presence of signal from two sensors that share a monitoring space, it's possible to glean more distance and bearing information than possible from a single sensor.

The cone defined by the intersection of range and bearing coverage for each sensor is defined by the presence of signals from both sensors (A&B). Similarly, the area covered by one sensor, exclusive of the area covered by the similarly facing sensor (A NOT B and B NOT A) can be computed, as well as the total area covered by both sensors (A OR B).

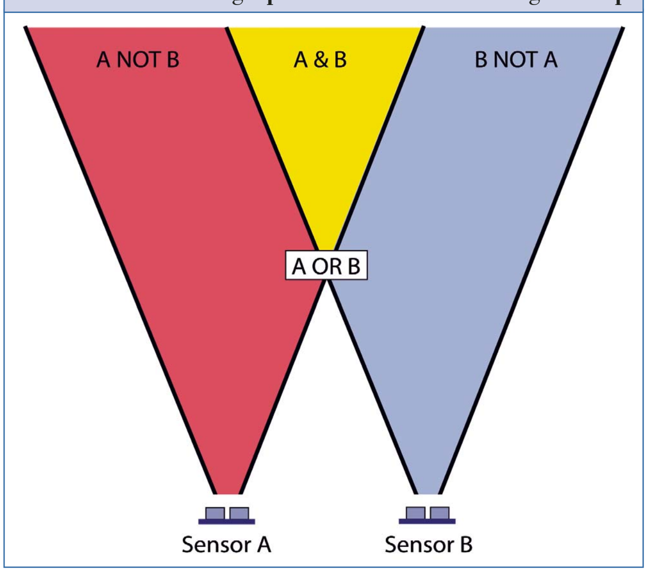
Using the presence or absence of raw sensor data from the Ping))) and GP2D12, a robot can locate and avoid both rubber balls and paint cans on the garage floor more accurately than with data from either sensor alone. However, differentiating balls from cans requires fusion at the data level, which involves distance measures. Assuming the sensors are arranged to provide overlapping coverage (as in Figure 10), the GP2D12 will find a rubber ball in the A&B area at a greater range than the Ping))), but the response would be reversed for a paint can target.

At the decision fusion level, how the sensor data are used to control overall robot behavior is dependent on sequencing and conditional use of data from each sensor. A robot can be programmed to respond to data from each sensor individually, and in a predefined order, as in Figure 11. In this popular configuration, the "other sensors" typically include a bumper switch.

Handling data from the GP2D12 before that of the Ping))) isn't significant for a slow-moving carpet rover. However, for a fast-moving robot, the choice of which sensor to read first can be critical. The option shown in Figure 11 is probably a poor choice for a garage filled with empty paint cans, because the range of the GP2D12 is considerably less than that of the Ping))). Furthermore, the cost — in terms of potential damage to the robot — is probably considerably greater for a robot-can encounter compared with a robot-ball collision.

One of many possible alternative fusion algorithms at the decision level is

FIGURE 10. Sensor logic possible with sensor coverage overlap.



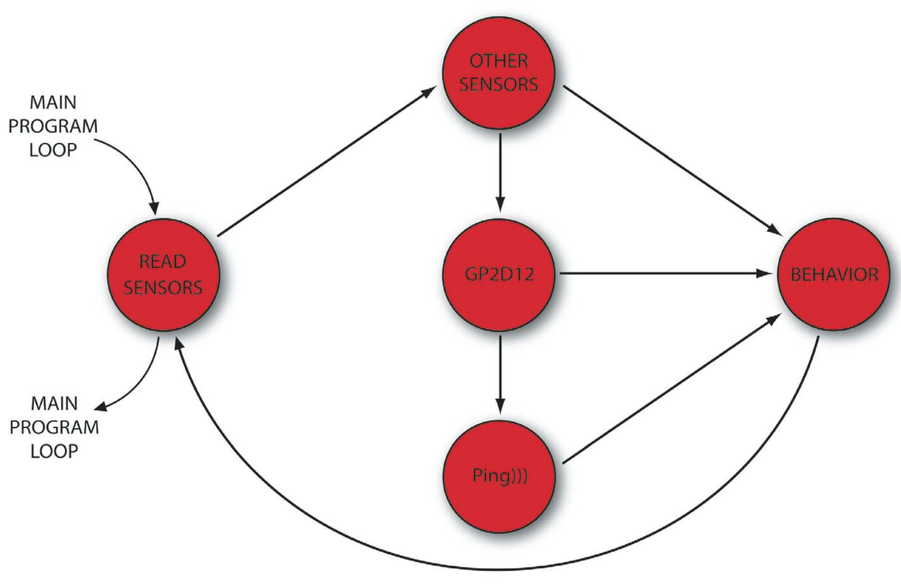


FIGURE 11. Decision fusion configuration in which each sensor is associated with the same behavior.

shown in Figure 12. In this example, Ping))) data are considered first. If there is no distance data from the ultrasonic rangefinder, then control is returned to the main program loop. This configuration allows power and time savings, because the additional sensors aren't fired at every cycle. This savings is at the expense of a higher likelihood of collision with rubber balls.

Conditional use of sensors is a common practice that, if used appropriately, can improve robot performance. For example, in Listing 1, the temperature is measured before every distance measurement. Unless the ambient temperature is highly dynamic, reading the temperature once during the initialization routine would be adequate. Additional decision configurations are possible, including different behaviors for each sensor. A battlebot might use a sharp weapon to deflate balls in its path, but choose to bludgeon the cans.

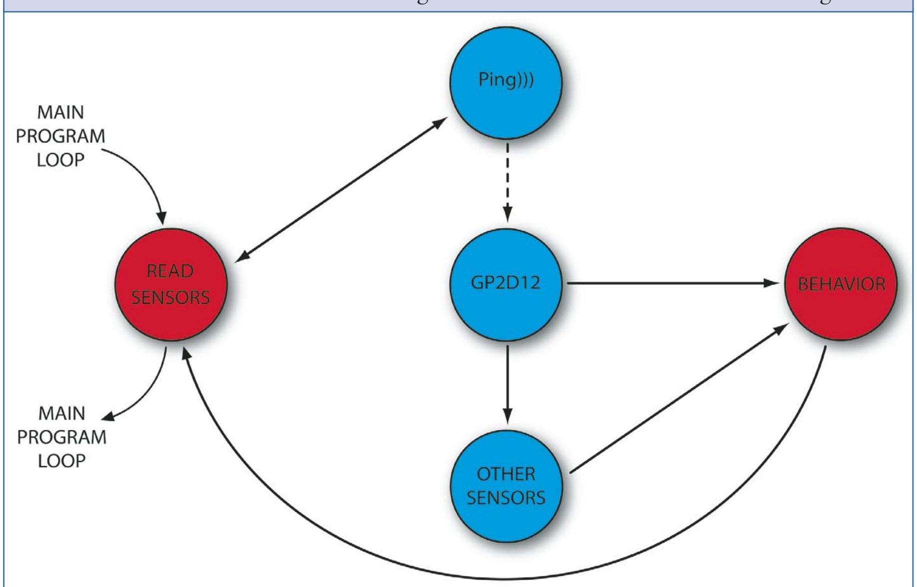
DISCUSSION

In working with these two sensors, there are issues of relative accuracy, maximum update frequency, cycle spacing, coverage area, environmental specificity, and failure rates. From the above discussion, it should be obvious that there is something to be gained from using data from both sensors from the signal level to the decision level. Fortunately, the science of sensor fusion is much richer than the limited discussion here.

A hexapod that has to avoid static obstacles on a garage floor can get by with a few hard-coded heuristics. However, when milliseconds count — as in a high-speed autonomous vehicle or missile locked on a target — then intuitive methods used here are inadequate. The effects of individual differences in sensors and seemingly minor changes in ambient temperature and supply voltage variations are magnified.

Furthermore, it's no longer sufficient to know that a target is up ahead somewhere — its current location and expected trajectory over the next few milliseconds become critical. Part 2 of this series extends the concept of sensor fusion to more powerful methods that function in dynamic targets and environments. **SV**

FIGURE 12. Decision fusion configuration with conditional sensor handling.



REFERENCES

[1] Braitenberg, Valentino., *Vehicles: Experiments in Synthetic Psychology*. 1984, Cambridge: MIT Press.

[2] Ping)))™ Ultrasonic Rangefinder (#28015). Parallax, Inc., June, 2005.

[3] Cramer, Olson., *The variation of the specific heat ratio and the speed of sound in air with temperature, pressure, humidity, and CO_2 concentration.* Journal of the Acoustic Society of America, 1993. 93(5) p(5): p. 2510-6.

[4] Allen, Tracy., *Temperature & Humidity With the Sensirion SHT1x.*, 2005, ***www.emesystems.com/OL2shtlx.htm***

[5] Williams, Jon., Measuring Up — Up to 80 Centimeters, That Is. *The Nuts & Volts of BASIC Stamps, Volume 5*. 2004: Parallax.

[6] uM-FPU Application Note 4 — Measuring Distance with the Sharp GP2D12 and GP2D120 Distance Sensors, 2005. Micromega Corporation. ***www.micromegacorp.com***

Building (H-)Bridges — Part 1 —

by Peter Best

Today's robotic creations use various methods — which are based on the physics of our little planet — to enable the motion of their main body or subordinate appendages. The various parts and pieces of these intelligent "things mechanical" use hydraulics, air pressure, muscle wire, and even gravity to invoke a mechanical displacement from Point A to Point B.

Despite the increasing use of the aforementioned physical methods in robotic equipment, the major component involved in making robotic things move is still the motor. Take a look at the advertisements and columns in *SERVO*. The majority of them have some sort of motor at their root. And, in most cases, if a motor is not in the column's or advertisement's mix, the electronics that drive or control a motor are.

Motors are the main motivation of this column, as well. However, instead of delving into the nuances of how motors use magnetic fields to create motion, I'm going to show you how to build electronic circuitry that controls the activation, deactivation, and direction of the electronic movement within a motor's magnetic domain.

Driving a Brushed DC Motor

Small brushed DC motors are fascinating. Their internal complexity is overshadowed by their ease-of-use. If the motor's minimum operating voltage is low enough, connecting the brushed motor's two power leads across a battery is all that is needed to get the motor's shaft to rotate. The fun comes in when you reverse the battery's polarity with respect to the motor's power leads. The motor shaft will then spin in the opposite direction.

If the brushed motor has exceptional bearings supporting the shaft, disconnecting the battery will result in the shaft coasting to a stop. The ability of a motor to provide forward motion, reverse motion, and to stop are physical properties used by every robotic device that employs the services of a traditional electromagnetic motor. However, to take robotic advantage of the work done by a motor, the motor's forward, reverse, and stop properties must be able to be controlled.

The circuit I've devised in Figure 1a is the most basic of brushed DC motor control designs. A positive voltage that is sufficient enough to turn on the MOSFET applied to the MOSFET's gate will provide a ground path for the motor through the MOSFET which, in turn, will put the motor's shaft into motion.

The Schottky diode that parallels the motor is there to allow a conduction path for the back EMF that is created by the motor coil when the motor shaft stops spinning. This type of motor control is great if all you want to do is drive the brushed DC motor full tilt in a single direction. If rotating the motor shaft in a single direction is fine, but you don't want your robotic device moving at warp speed all of the time, you must consider controlling the speed of the brushed DC motor.

With the circuit shown in Figure 1a, speed control is easily achieved by simply applying a PWM (Pulse Width Modulation) signal to the MOSFET's gate. The higher the on time (logical high) of the PWM signal, the faster the motor's shaft will spin.

What if your little robotic device had to use a brushed DC motor and controller in the Figure 1a configuration to move one of its mechanical parts from Point A to Point B and then return to Point A? I'm thinking about some really nasty things that have to do with DPDT mechanical switches to switch the brushed DC motor's

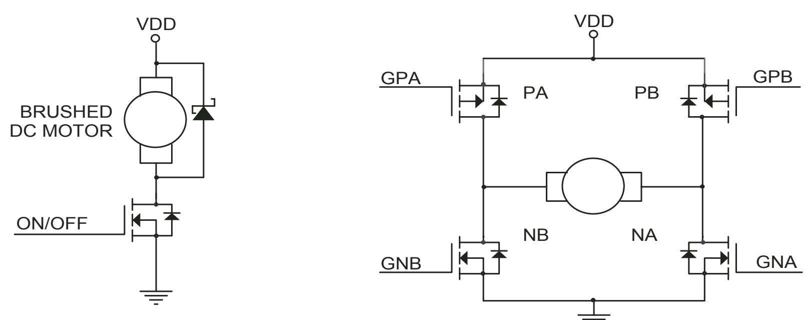
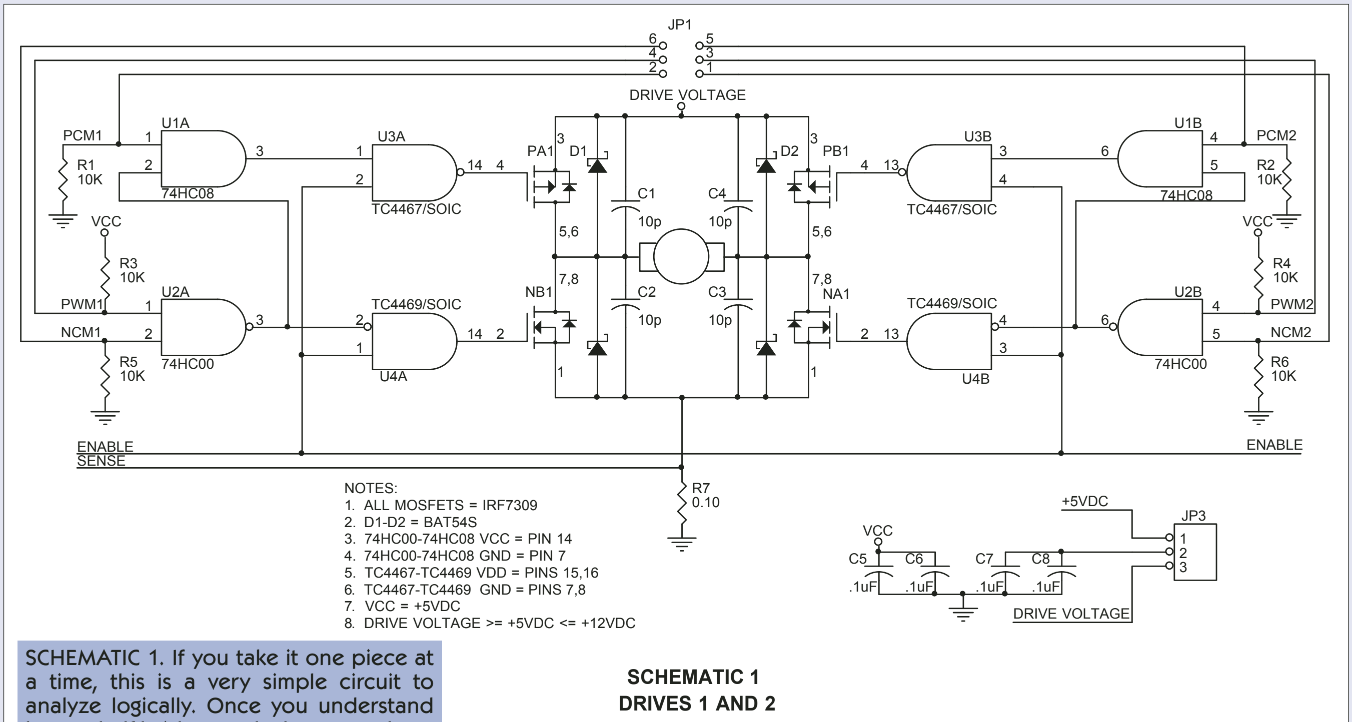


FIGURE 1A

FIGURE 1B

GPA	GPB	GNA	GNB	FUNCTION
1	1	0	0	OFF (FREEWHEEL)
0	1	1	0	FORWARD
1	0	0	1	REVERSE
0	1	0	1	SMOKE
1	0	1	0	SMOKE

FIGURE 1. (A) This is Electronics 101 stuff. If you want to see the magic smoke, just leave the Schottky diode out of this little circuit. Also, if you want to experiment with this circuit, be sure to place a 100 ohm resistor in series with the MOSFET gate. (B) Don't try this at home! This circuit is bare bones and is for illustration purposes only. Although the MOSFETs will switch and drive the motor per the truth table, there is no protection for the MOSFETs in this circuit other than their internal diodes.



SCHEMATIC 1. If you take it one piece at a time, this is a very simple circuit to analyze logically. Once you understand how a half-bridge works here, you have the key to understanding what happens when you combine half-bridges to form full H-Bridges.

power terminals into reverse mode that we really don't want to discuss. So, let's add a reverse gear to our brushed DC motor electronically.

In order to switch the brushed DC motor's power leads between being sourced and sinked, you need the circuit in Figure 1b, which adds a pair of P-Channel MOSFETs (PA and PB) to provide the sourcing of power to each of the brushed DC motor's power leads. The sinking function is provided by a pair of N-Channel MOSFETs — NA and NB — which are tied to each of the brushed DC motor's power leads, as well.

Note that one N-Channel MOSFET drain and one P-Channel MOSFET drain are connected to each of the brushed DC motor's power leads. Power to the brushed DC motor enters at the P-Channel MOSFETs' source pins. The brushed DC motor's power path to ground is provided by the sinking N-Channel MOSFETs — NA and NB.

For the sake of discussion, let's assume that MOSFETs PA and NA associate with clockwise rotation of the brushed DC motor shaft and PB and NB associate with counter-clockwise rotation of the brushed DC motor shaft. With that, to spin the motor shaft in a clockwise direction, we must provide (source) power to the motor using the PA MOSFET. The other motor power lead must somehow get to ground. The N-Channel MOSFET — NA — provides for a ground path for the motor and performs the sinking, or grounding, function upon its activation. We can apply the same logic for counter-clockwise rotation of the motor shaft by activating MOSFETs PB and NB. I've assembled a motor shaft direction truth table with Figures 1a and1b.

Notice in the Figure 1 truth table that a couple of combinations of activated MOSFETs result in SMOKE. Never do we want to activate MOSFET pairs PA and NB or PB and NA at the same time. It's pretty obvious in Figure 1b that activating these pairs in any combination will produce a path from the source voltage to ground through the MOSFETs, which have very low drain-to-source resistances.

In other words, activating the PA/NB and/or PB/NA MOSFET pairs will produce a virtual short circuit from the power source to ground. If your coding prowess is exceptional and you feel that you can handle switching the MOSFET pairs correctly using only your firmware, go for it. However, some simple hardware placed in front of the MOSFET pairs will assure that your "perfect" firmware won't take the basic H-Bridge shown in Figure 1b into SMOKE mode.

Controlling the H-Bridge

Our main objective here is to provide a fool-proof control mechanism for the MOSFET pairs that make up our H-Bridge. Rather than depend on "perfect coding," let's employ the services of a couple of MOSFET driver ICs that bring a bit more to the table than just being able to drive a MOSFET gate.

The Microchip TC4467 and TC4469 are four-output CMOS buffers/MOSFET drivers that can, by themselves, deliver up to 1.2A of peak drive current. In fact, you can actually drive motors that require less than 250 mA of current directly from the TC4467 or TC4469 output pins.

The difference in the TC4467 and TC4469, when compared to other MOSFET drivers, is their inclusion of integral logic gates to complement the MOSFET drivers. Each of the four output drivers in both the TC4467 and TC4469 is front-ended by a two-input logic gate. The TC4467's input pair

feeds a standard NAND gate while the TC4469 logic gates are configured as AND with an inverted input.

I've added a 74HC00 and 74HC08 to the TC4467 and TC4469 H-Bridge controller mix in Schematic 1. Be aware that although the components in Schematic 1 look to be in a full H-Bridge configuration, they are not. What you actually see in Schematic 1 is a pair of half-bridges. PA1 and NB1 make up one of the half-bridges and PB1 and NA1 comprise the other half-bridge. Placing jumpers across JP1 will combine the pair of half-bridges into a full H-Bridge. For now, we'll forego the JP1 jumpers and keep the configuration in half-bridge mode as I walk us through the bridge control logic.

The idea is to not allow the vertical pairs of MOSFETs to be activated simultaneously. So, let's logically analyze the logic gates to see if our protection circuitry works. PCM (P-Channel MOSFET) 1/2, PWM (Pulse Width Modulation) 1/2, and NCM (N-Channel MOSFET) 1/2 are all bridge control inputs. Let's look at what the PA1 and NB1 MOSFET gates look like logically with no input stimulus on the PCM1, PWM1, and NCM1 inputs. Assume the ENABLE line to be active, or logically high.

Since pin 1 of U1A is pulled low, the output of AND gate U1A will be logically low regardless of the logic level applied to U1A pin 2. With the ENABLE line held at a logically high level, the inputs currently being applied to the TC4467 NAND driver will produce a high level on the output of U3A, which turns PA1 off. With one input pulled high and the other pulled low, U2A's output pin will be logically high.

The high level on the output of U2A feeds the invert input pin on the TC4469 MOSFET driver. Thus, pin 2 of TC4469 is effectively a logical low input. Since U4A is an AND gate with an inverted input, the output of U4A will be low, turning off MOSFET NB1. The PB1/NA1 half-bridge circuitry is identical and so are the results of logic levels applied to the gates of PB1 and NA1 with relation to input levels applied to PCM2 and NCM2.

PCM1, in conjunction with the ENABLE input, is used to turn on MOSFET PA1. Introducing a logically high level to pin 1 of U1A with no external input stimulus applied to U2A results in both of U1A's input pins presenting a high logic level to the 74HC08 AND gate, which produces a high at pin 1 of U3A. With the ENABLE providing a high input level at pin 2 of U3A, the TC4467 NAND gate's output goes logically low to turn on MOSFET PA1.

Here's where things get interesting. NCM1 is used to turn on MOSFET NB1. Let's assume PA1 is on and we apply a logical high to the NCM1 input to turn on NB1. Again, ENABLE is active and is held in a logical high state. Taking NCM1 logically high produces a logical low on the output pin of the NAND gate U2A, which feeds a low to the input of AND gate U1A, which drives the output pin of U1A low, which drives the output pin of NAND gate U3A high and turns off PA1.

Meanwhile, the low level on the output pin of NAND gate U2A feeds the inverted input pin of the TC4469 AND gate, which results in a high being fed to the gate of NB1 turning the N-Channel MOSFET on. Once again, the control input logic that works for PA1 and NB1 works identically for PB1 and NA1.

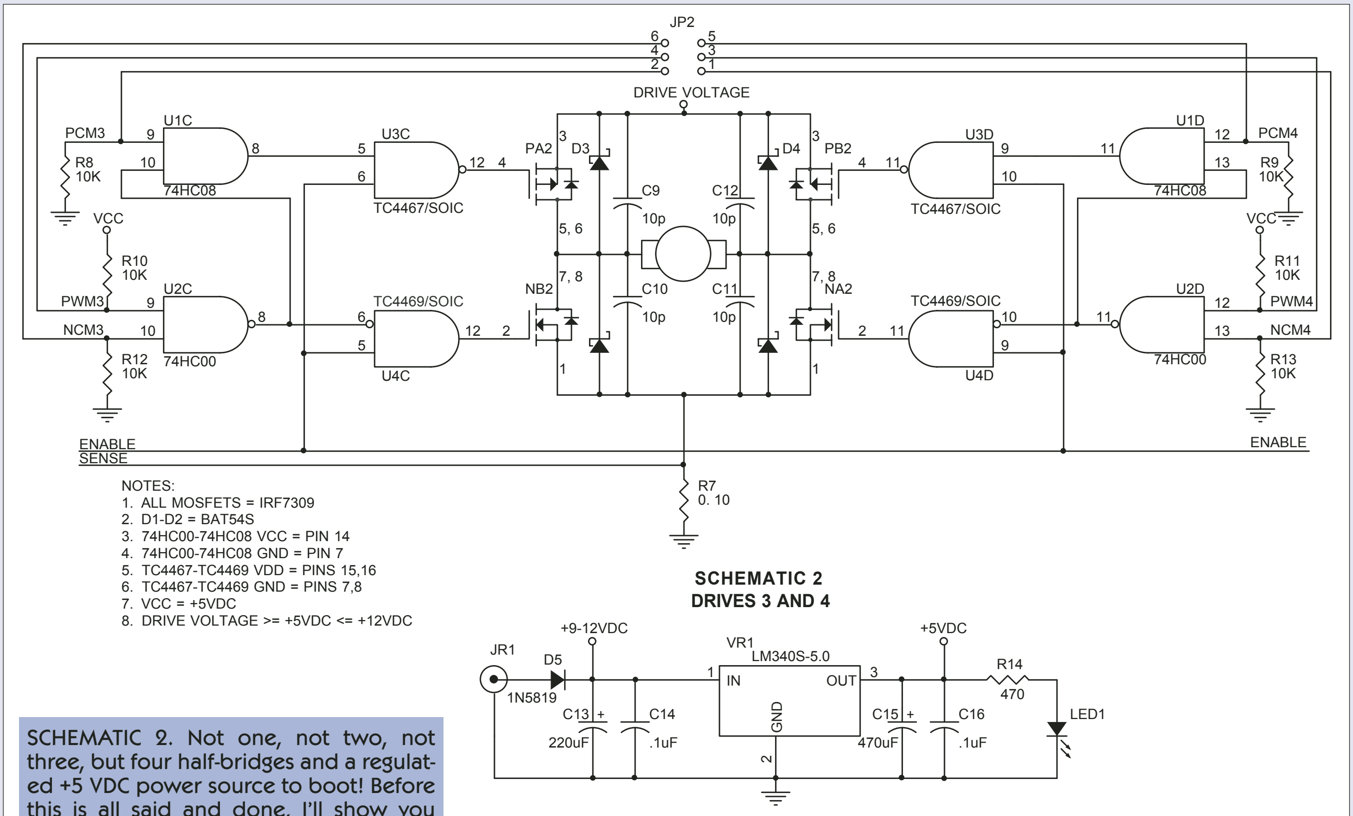
Okay, our MOSFET protection circuitry works great on paper. Let's check out the logic again and make sure we can actually turn the brushed DC motor shaft in both directions using half-bridge mode. Earlier, we associated PA1 and NA1 with clockwise rotation. So, to initiate clockwise motor shaft operation, we must energize PA1 and NA1. Applying a logical high to the PCM1 input will turn on PA1, which will allow the DRIVE VOLTAGE to flow to one of the brushed DC motor power leads.

NA1 is activated by issuing a logical high level to the NCM2 input. Once NA1 is on, the brushed DC motor's ground path is established causing the brushed DC motor's shaft to turn in a clockwise direction. To shift the brushed DC motor's shaft into reverse, we must remove the input stimulus from PCM1 and NCM2 and apply logical high levels to PCM2 and NCM1. The removal of clockwise stimulus before applying the counter-clockwise logic levels is necessary because we are actually controlling an independent pair of half-bridges.

If we were to jumper pins 1 to 2, 3 to 4, and 5 to 6, applying a logical high to PMC1 or NCM2 would turn on PA1 and NA1 and result in clockwise rotation of the brushed DC motor's shaft. Applying a logical high to the PCM2 input with the JP1 jumpers in place would result in turning on PB1 and NB1 and counter-clockwise rotation of the brushed DC motor's shaft. Thus, with the JP1 jumpers populated, we combine the pair of half-bridges into a full H-Bridge with all of the safety features we designed still intact.

In our simplified half-bridge brushed DC motor driver scenario, the PWM signal is applied to the NCM1 or NCM2 inputs. Since PWM1 and PWM2 are both tied logically high, the alternating PWM signal presented to the NCM1 or NCM2 inputs will force the MOSFET gates of NA1 or NB1 to chop the brushed DC motor's ground path relative to the duty cycle of the incoming PWM signal and thus,

"Today's robotic creations use various methods — which are based on the physics of our little planet — to enable the motion of their main body or subordinate appendages."



SCHEMATIC 2. Not one, not two, not three, but four half-bridges and a regulated +5 VDC power source to boot! Before this is all said and done, I'll show you how to put all of these half-bridges to work in various configurations.

regulate the motor shaft's speed.

In most real-world cases, the PWM signal emanated by the microcontroller will be continuous as the programmer will simply kick off the microcontroller's PWM engine and only manipulate the PWM duty cycle as needed. With the PWM running continuously, most likely a full H-Bridge configuration will be used and the NCM1 or NCM2 inputs are then used as gates to allow or disallow the PWM signal from passing on to the N-Channel MOSFET gates.

Schematic 2 utilizes the rest of the gate logic inside all of the ICs that make up the four half-bridges. The TC4467 and TC4469 can be powered by the standard logic supply of +5 VDC or by the DRIVE VOLTAGE, which can span from +5 VDC to +18 VDC. A jumper on JP3 determines which supply the TC4467 and TC4469 draw their power from. If you're scratching your head as to why I haven't mentioned the 0.10 ohm sense resistor, don't worry. I've got plans for that guy.

The H-Bridge Hardware

Enough theory ... let's build something! I've assembled all of the components in Schematics 1 and 2 into the dual H-Bridge hardware you see in Photo 1. The printed circuit board itself is an inexpensive double-sided board with a ground plane covering the entire non-component side. There are so few components that comprise the dual H-Bridge circuit that you can actually build this whole thing up by looking at the component placement in Photo 1.

The H-Bridge circuitry you see in Photo 1 can be more easily understood when broken down into its component parts. Photo 2 details the anti-smoke logic and the TC4467/TC4469 MOSFET drivers. From left top to right

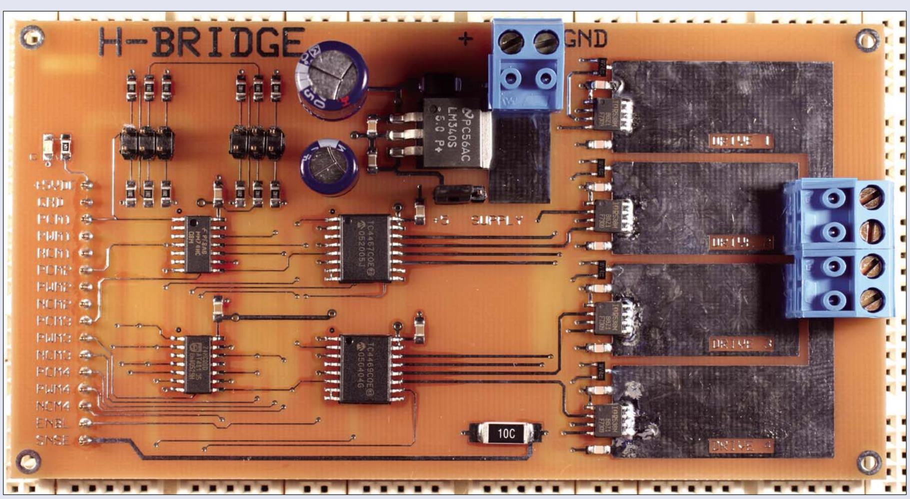


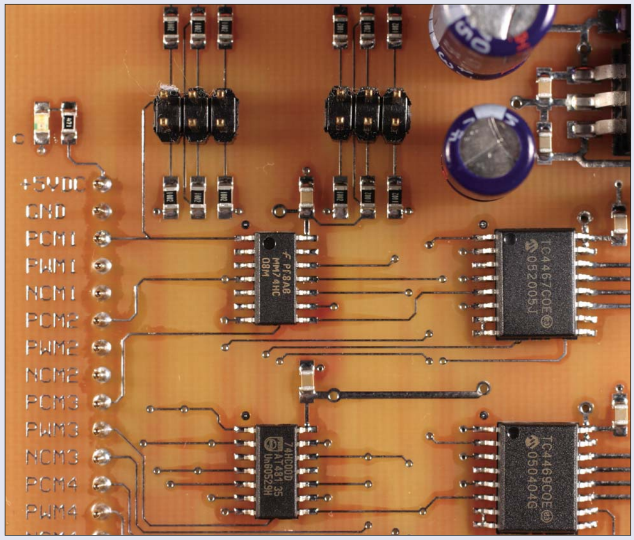
PHOTO 1. There are actually four half-bridges buried within this mix of logic and MOSFET drivers. You can run this baby as four independent half-bridges or a pair of full H-Bridges. In addition to being able to drive small motors with ease, the circuitry boasts some simple logic that prevents you from accidentally letting the magic smoke out of the MOSFETs and their drivers.

top, you see the power indicator LED and its respective current limiting resistor. Directly to the right of the power indicator LED/resistor combination is JP1 which, when jumpered, puts half-bridges 1 and 2 (or Drives 1 and 2) into a single, full H-Bridge configuration. JP2 — whose jumpers put Drives 3 and 4 into full H-Bridge mode — is positioned to the right of JP1.

Photo 3 is a bird's-eye-view of the H-Bridge +5 VDC logic power supply. There's nothing fancy here as the dual H-Bridge logic power supply is constructed around an LM340S-5.0 fixed voltage regulator. The logic power supply always powers the 74HC00 and the 74HC08 and can be jumpered to supply +5 VDC power to the MOSFETs, TC4467 and TC4469. The Drive Voltage jumper — which is just below the logic power supply — can also be positioned to provide the voltage at the input of the LM340S-5.0 to the MOSFETs, TC4467 and TC4469. The Drive Voltage jumper is set across the +5 VDC position in Photo 3.

The 10 pF EMI (Electromagnetic Interference) capacitors, the BAT54S made up of a pair of back-EMF steering diodes, and the IRF7309 MOSFET pair are all packed into the electronic components you're close up on in Photo 4. All of the IRF7309 MOSFET drains (pins 5, 6, 7, and 8) are tied together and bonded to a one-square-inch heatsink/drive output pad. The heatsink size is directly proportional to the amount of current you want to pull through the bridge's MOSFETs.

PHOTO 2. There are 16 logic gates and eight MOSFET drivers in this shot. Note the bridge control inputs pinned out to the left and the bridge configuration jumper blocks, which are bare indicating half-bridge mode. The dozen 10K resistors surrounding the bridge configuration jumper blocks guarantee that the MOSFETs are turned off in the absence of input stimulus at the bridge control inputs.



Crossing the Bridge

I've provided the dual H-Bridge ExpressPCB layout file — which is available on the *SERVO* website **(www.servomagazine.com)** — for those of you that may wish to customize the H-Bridge design I've presented or simply order your own set of dual H-Bridge boards from ExpressPCB directly. For those of you that are surface mount challenged, you can also get all of the SMT components that make up the dual H-Bridge I've described as through-hole and DIP packaging, with the exception of the IRF7309 MOSFETs. There's nothing to stop you from using MOSFETs that are packaged differently than the IRF7309s.

If you haven't skipped through the pages to get here, you've made it to the end of this part of our H-Bridge discussion. You now know how to control each and every one of the MOSFETs on all four half-bridges and, once you build up the H-Bridge circuit I've offered to you, you can attach a brushed DC motor and jumper in the bridge control logic levels necessary to make the motor shaft spin in a clockwise or counter-clockwise direction. If you already have the means and knowledge to do so, you can also present a PWM signal to the proper bridge control inputs and control the speed of your motor's shaft.

Although we've crossed the "bridge," we're not done. Next time, I'll introduce you to the brand new PIC16FHV616 and show you how to use simple PIC assembler code to exploit all of the PIC16HV616's special functionality to drive stepper and brushed DC motors with the H-Bridge circuitry you've just read about. I'll also produce some circuitry and firmware to put that 0.10 ohm sense resistor to work as a vital component of a safety net and motor current monitor station for the H-Bridge MOSFETs. **SV**

Peter Best can be contacted via email at peterbest@cfl.rr.com

PHOTO 3. Imagine that ... a National Semiconductor fixed voltage regulator providing regulated power to the logic. The screw terminal in this shot is the portal for incoming power for the dual H-Bridge logic and motor.

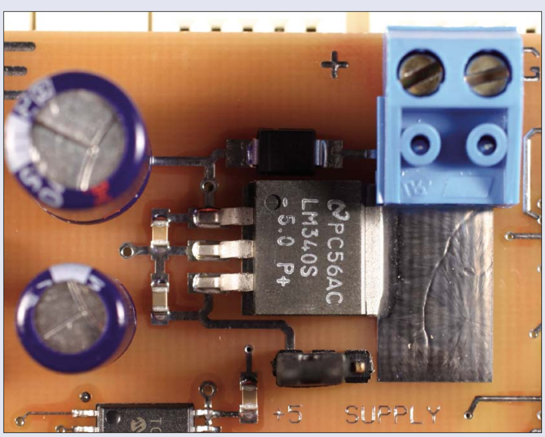


PHOTO 4. Here's a complete half-bridge, including the back-EMF steering diodes and the EMI capacitors. The IRF7309 was designed to switch things about in laptop computers and other electronic devices that require higher switching currents from smaller footprint components.

by Jack Buffington

Beyond Bleeps and Bloops

How to Record and Play Back Any Sound

In the past, this column has shown you how to generate sounds through synthesis and how to generate speech from plain old ASCII text. This month, you'll see how easy it is to add audio recording and playback capability to your robot. If you have ever wanted to have your robot play old sci-fi sound effects or possibly cry out for help when it encounters a situation that it can't deal with, then this is the column for you!

Play It Again and Again

A company called Winbond produces chips that let you record and play back audio. With the chips in their ChipCorder line, you will be able to record up to 16 minutes of audio for later playback. Many of these chips have been available for years and are readily available from places such as Digi-Key. These chips are capable of producing audio at frequencies of up to 3.4 kHz (with a sample rate of 8 kHz) so they won't be able to produce high-pitched sounds, but will do fine with most sounds that you would like to play back.

Meet the ISD4003-04MP

The chip that this column will discuss is the ISD4003-04MP. It can play back up to four minutes of audio with a sample rate of 8 kHz. It has 28 pins and is packaged in a DIP package that is .6" wide. The ISD4003 requires very few external parts in order to record and play back.

For the most part, all you will need are a few capacitors. This chip has an internal clock to control the speed of its recording and playback that is accurate to within -6% to +4% of the needed frequency over a large temperature range, but if you require higher precision or simply need your audio to remain in perfect sync with your application, then you can provide an external clock source.

You can communicate with the ISD4003 using only four pins. These pins are a slave select pin and three SPI (Serial Peripheral Interface) pins. There are two other pins that you might be interested in using. These pins are the 'row address clock' pin and the 'interrupt' pin. Internally, the ISD chip has a large 2D array of analog Flash cells. This is something of a novel technology since most Flash devices store only high and low values that correspond to a one or a zero.

Internally, the ISD chip doesn't use compression to be able to store all of the sound, it just writes the sounds directly to the analog Flash. The ISD's Flash memory is arranged into rows and columns. During playback, every time that it reaches the end of a row, the row address clock pin will drop low for 25 milliseconds. On a chip that has a sample rate of 8,000 Hz, these interrupts happen every 1/5th of a second.

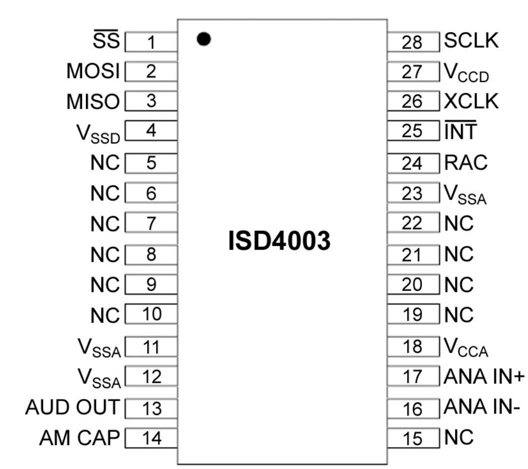
Pin Interrupted

The other pin that you might be interested in is the interrupt pin. This pin will drop low when the end of a recording is reached during playback. It will also go low during recording if you fill up the memory of the chip. Using the interrupt pin allows you to do other things while the chip is playing. If you want to know when a sound is done playing, you can monitor the interrupt pin, otherwise you will need to monitor the row address clock or just delay for the amount of time that it takes to play your sound.

Chip Basics

Let's get to the basics of how this chip works. As mentioned earlier, this chip communicates with an external processor using SPI. SPI uses two data lines and a clock line. Typically, a

Figure 1. The pinout of the ISD4003 chip.



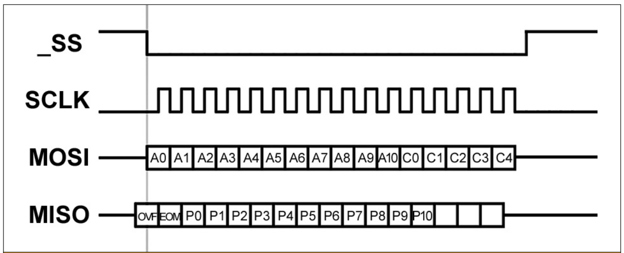


Figure 2. The data that is sent and received from the ISD chip.

fourth line is also used. This line is usually called Chip Select (CS) or Slave Select (SS). The two data lines are usually labeled MOSI and MISO. These stand for Master Out Slave In and Master In Slave Out. The master is the chip that generates the clock signal.

In essence, SPI hardware is really just two shift registers. One that data is clocked out of and one that data is clocked into. The register that shifts data to the master latches its data when the slave select line drops low, and the latch that receives data from the master latches its data when the slave select line goes high.

To use the SPI port on the ISD4003 chip, you will need to do the following: When you want to communicate with the ISD chip, you should drive the SS line low. This tells the chip that you are about to communicate with it. You will now start driving your clock line high and low. You need to write your data to the MOSI line when the clock is low and read data from the MISO line when the clock is high.

When you are done sending and receiving data from the chip, you will drive the slave select line high. All commands to the ISD are 16 bits in length. This is convenient if your processor happens to have a hardware SPI port on it. With two writes to the SPI port, you can send one command and leave your processor free to do other things. If you don't have an SPI port, then communication is still pretty easy to do with software.

Figure 2 shows a diagram of the signals needed to communicate with an ISD chip. As you can see, there are 11 address bits and five control bits in each packet. The address bits are sent first and specify which row you would like to set the playback pointer to. There are 1,200 rows inside of each ISD4003 chip, so addresses of zero to 1,199 are possible. Figure 3 shows the list of commands that you can send to the ISD4003.

Putting the ISD4003 to Use

You now know all of the commands and how to send them. Let's look at what you need to do to actually use this chip. All of the timing described here is for a chip that has a 8,000 Hz sample rate. There are some odd delays that you will need to take into account when using this chip. To play back when the chip is powered down, you will need to send the 'powerup' command and then wait 25 milliseconds. Send the 'setplay' command with the address that you want playback to start from and then send the 'play' command. You will now hear whatever you have recorded at that location. The ISD chip will continue to play until it gets to the end of the recording or the end of memory in the chip. If you want to, you can send the 'stop' command to stop the playback.

Recording is done a little differently. You will need to send the 'powerup' command twice to enter recording mode. The first time, you will need to wait 25 milliseconds. After the second time you send the 'powerup' command, you will need to wait 50 milliseconds. Now send the 'setrec' command with the address that you would like to start recording from. Next, send the 'rec' command to start recording. When you are done recording, send the 'stop' command.

As you can see, knowing the address is important for starting and stopping playback and recording. You can figure out the address by sending any command. At the same time that you are sending the command, the ISD chip will send two status bits corresponding to if playback or recording has hit the end of the memory (OVF) or the end of the recording (EOM).

After these two bits will be the current address that playback is happening from. The remaining three bits don't mean anything.

Command	Address (11 bits)	Control Bits C0 C1 C2 C3 C4	Description
POWERUP	Doesn't matter	0 0 1 0 0	Power up: Device will be ready for an operation after T_{PUD} (power up delay).
SETPLAY	Address (0-10)	0 0 1 1 1	Plays from the address given.
PLAY	Doesn't matter	0 1 1 1 1	Plays from the current location.
SETREC	Address (0-10)	0 0 1 0 1	Records from the address given.
REC	Doesn't matter	0 1 1 0 1	Records from the current location.
SETMC	Address (0-10)	1 0 1 1 1	Initiates message cueing from the address given.
MC	Doesn't matter	1 1 1 1 1	Performs message cueing from the current location. Proceeds to the end of the message.
STOP	Doesn't matter	0 1 1 X 0	Stops the current operation.
STOPPWRDN	Doesn't matter	X 1 0 X 0	Stops the current operation and enters into standby (power down) mode.
RINT	Doesn't matter	0 1 1 X 0	Read the interrupt status bits: Overflow and EOM.

Figure 3. The messages that can be sent to the ISD4003 chip.

The 'rint' command allows you to read the address and interrupt bits without affecting the state of the chip.

Great! You are now almost ready to actually use this chip. The only thing left to do is to actually hook up the chip. Figure 4 shows the connections that you will need to make to play audio. Make sure to use a 3V regulator with this chip. This is the only exotic part that you will need. You can make a 3V regulator using a LM317 voltage regulator and a few extra parts or by having a few diodes in series with the 5V regulator that you are likely to be using with your processor.

The 3V regulator is not absolutely necessary. These chips will run at 5V, but you will have better audio quality at the three volts that the data sheet recommends. The SPI pins are five-volt tolerant when running at 3V, so interfacing with a processor that is running at 5V isn't a problem.

The ISD chip has a small amplifier inside of it but if you need some real volume, you will need to connect an amplifier to its output. Figure 5 shows a circuit that amplifies the output. To get the audio data into the chip, you will need to connect things as shown in Figure 6. Using that circuit allows you to record from a source such as your computer's audio out connector.

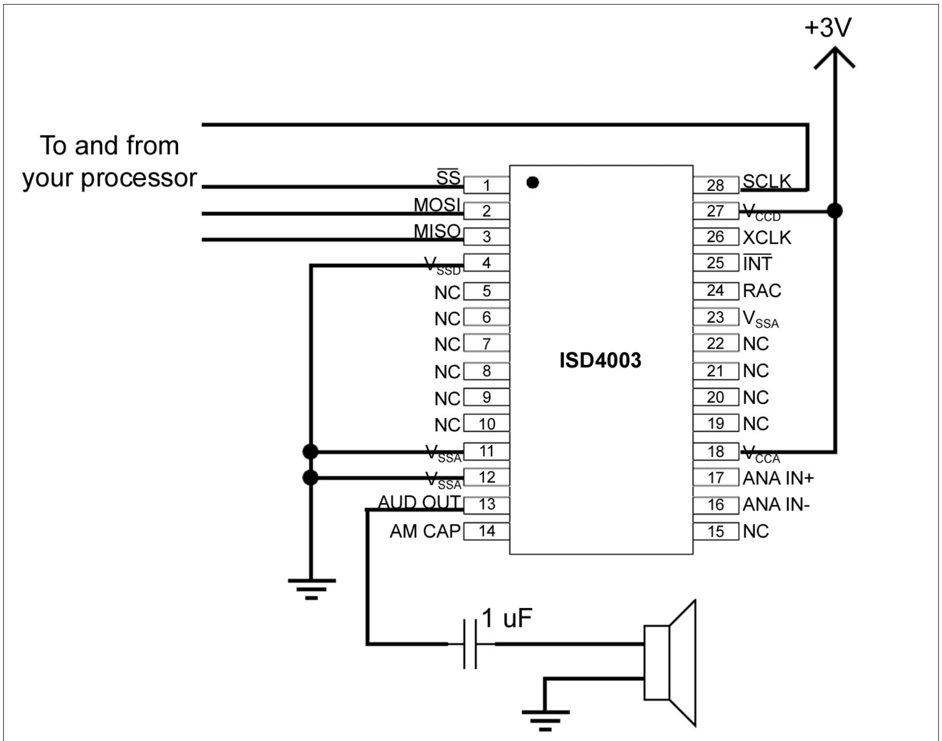


Figure 4. The connections that you will need to play sounds.

Wrapping It Up

Adding the ability to play around with recorded audio can let you do fun things like make a robot that slowly sneaks around where you live and occasionally lets loose with some scratching sounds to tease your cats. You could add sound effects for various things. For example, when your robot backs up, it could beep like a large truck does. You could also have it play knocking sounds when it reaches a door. If you were really ambitious, you could record a bunch of individual words and give your robot a voice — your voice! What will you do? **SV**

Figure 5. Here's how to amplify the audio.

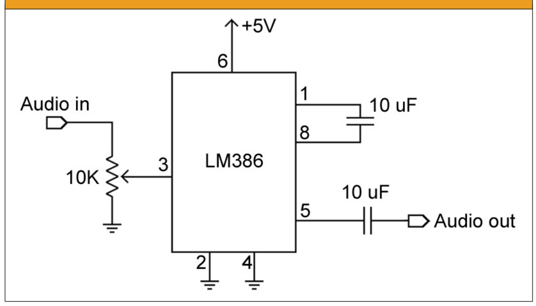
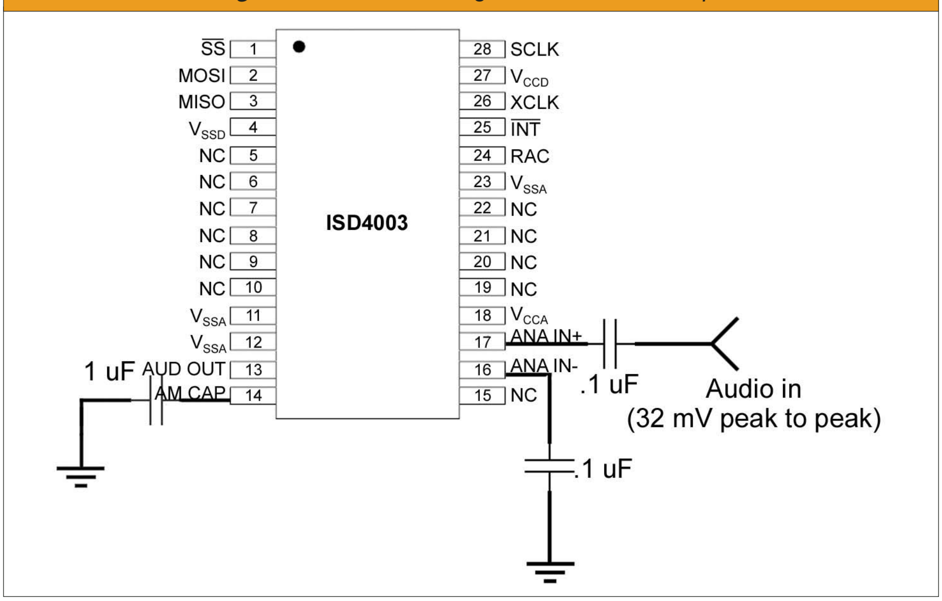


Figure 6. Here's how to get audio into the chip.



RESOURCES

Digi-Key
www.digikey.com
Sells various chips by ISD.

Winbond
www.winbond.com/e-winbond htm/partner/_ISD_VIC_CC.htm
Information about the various ISD chips.

Programmable Logic

by Gerard Fonte

The goal of this bimonthly column is to provide a basic understanding of the various programmable logic techniques.

There are a lot of powerful low-cost components available today that are rarely considered by hobbyists — and even some engineers — because of unfamiliarity.

You have to be comfortable with the idea and concepts of programmable logic before you will be likely to employ them.

Imagine a system where you sit at your computer and create a complicated digital circuit. Some simulations are run and they look good. You then press a button and the circuit becomes real. Such systems exist today and are actually very common. They use Field Programmable Gate Arrays (FPGAs), which are sometimes referred to as ASICs (Application Specific Integrated Circuits). These chips can replace a whole printed circuit board (or whole systems) of standard digital ICs (Integrated Circuits), with prices starting at around $10 to $15.

Xilinx vs. CPLD

We ended the last session with CPLDs (Complex Programmable Logic Devices). The Invert/AND/OR structure was shown to be flexible and powerful. However, there are problems with that approach. The first is that the number of Invert/AND/OR gates increases directly with the number of input and feedback signals. There has to be one for each input term. But, obviously, not all input signals and all feedback signals are used for every logic function.

On the average, only a very small percentage of these signals will be used for any given function. This means that there will be a significant amount of waste. In fact, it is not uncommon to use only 10% of the available Invert/AND/OR gates on a CPLD. Since ICs are basically priced by the area of silicon used plus a fixed package cost, this makes larger and larger CPLDs more and more cost *in*effective.

The second problem was the problem of feedback from the output registers. While this was possible, it generally wasted an input/output pin. Clearly, this is not efficient either. And if your design required a number of counters, you easily run out of available pins quickly. This isn't very good as well.

Enter the FPGA. In 1986, the Xilinx Corporation developed a decidedly new approach. They buried a large number of very small PALs on a single chip with programmable interconnects. This allowed the direct connection of any point to any other point (discussed in further detail later) and eliminated the inefficient Invert/AND/OR matrix.

For the actual PAL logic, they used memory logic (as described in this column in the March '06 issue of *SERVO*). This created identical delays regardless of the complexity of the logic. Finally, they took the programming off-chip. You didn't program the chip — you programmed a standard memory IC (or a special one of theirs). At start-up, the Xilinx chip would automatically download the design from the memory into on-board SRAM (Static Random Access Memory), which took a fraction of a second, then disable the external memory and then proceed to operate as designed.

This was a really big change. The off-board memory was typically an EPROM (Eraseable Programmable Read Only Memory) that was extremely common. So, if there was a problem with the design, all you had to do was to erase the EPROM, fix the design, and reprogram the EPROM with the new version. Hardware hadn't just become as easy to change as software. Hardware became software. It was the first large scale, re-programmable logic device.

This characteristic can be very powerful. For example, your new Internet appliance can also be upgraded via the Internet. Or, if your robot is acting up at a competition, you can

phone home for a hardware fix. Consider the possibilities. (Note that the removal of power automatically erases the on-board SRAM. The design must be re-loaded every time power is applied.)

There are a number of different manufacturers of FPGAs today (see the References sidebar). Xilinx is the front runner — Altera, Lattice, Actel, QuickLogic, and others round out the field. Each has its own approach that they think is best. However, Actel and QuickLogic use what is called "anti-fuse" technology. This means that the devices are not re-programmable and maintain their design in the power-off state. Since FPGAs start at $10, few hobbyists will find it financially feasible to spend $10 every time they want to change their design, especially when re-programmable solutions are available.

Since the FPGA architecture varies significantly from manufacturer to manufacturer, it is not possible to provide generic examples as was done with PALs and CPLDs. However, all FPGAs still have to perform the same three basic functions: logic, interconnection, and input/output. We'll use Xilinx examples for these (a somewhat arbitrary choice) with the understanding that other manufacturers perform these functions differently.

Basic Xilinx Architecture

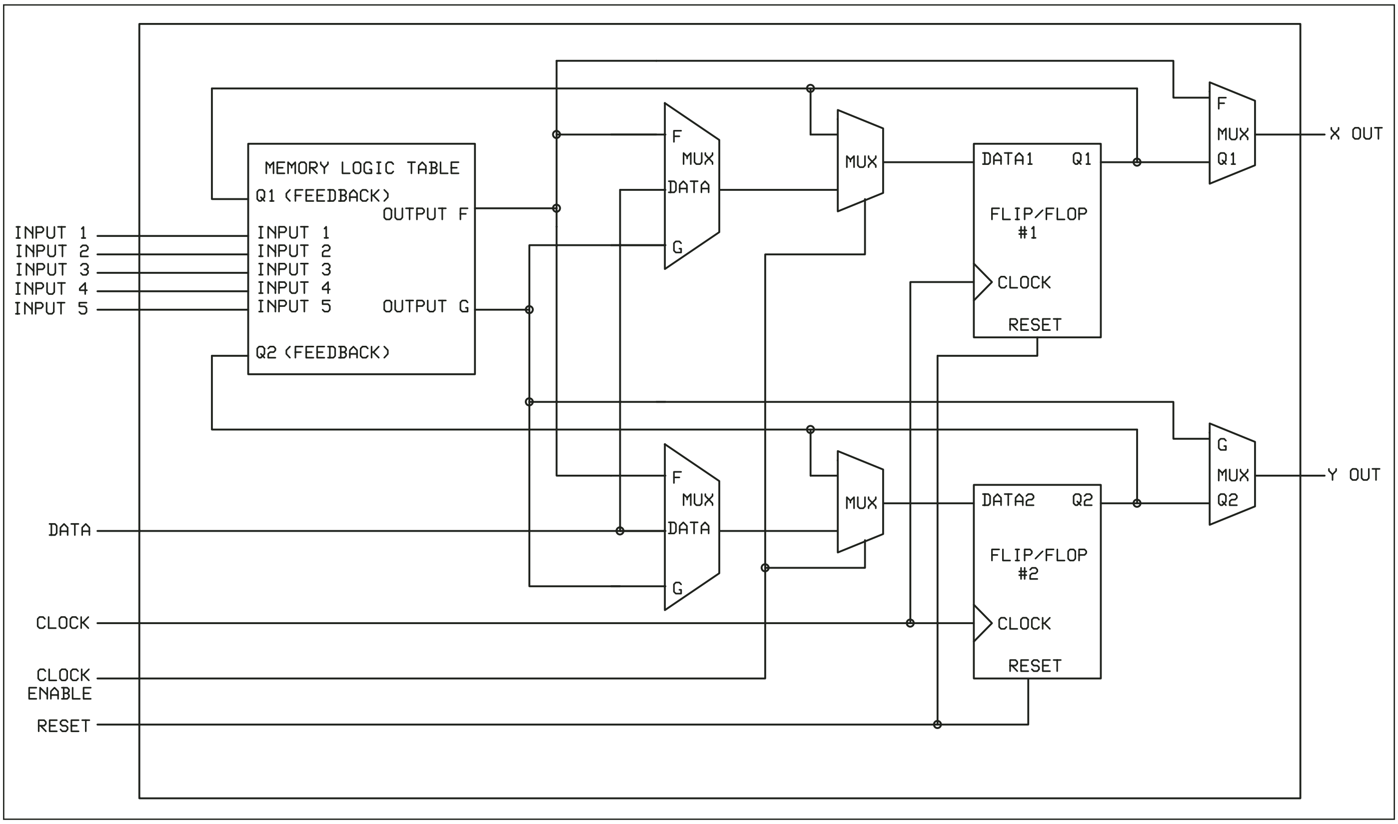
Figure 1 shows the basic Configurable Logic Block (CLB) for a Xilinx 3000 series FPGA. It's somewhat simplified and there are special functions and limitations that will not be discussed. In general, you can see that it resembles a small, registered-output PAL (Programmable Array Logic) that was discussed last time. The PAL's Invert/AND/OR logic has been replaced with a memory look-up table. There are three groups of signals: logic inputs, control inputs, and I/O (Input/Output). The logic inputs are self-explanatory. The control inputs are relatively more extensive than with a typical PAL or CPLD (Complex Programmable Logic Device). The outputs — like the registered PALs — can be driven from the logic look-up table directly or from the D-type flip-flops. These CLBs make up the core logic of a XIlinx IC and number from 64 to thousands on a single chip.

Interconnecting these CLBs are "Routing Resources." These are strips of silicon that act like wires. Transistors are turned on to make a connection. There are "local" resources that connect to adjacent CLBs and there are "Long Lines" that run the length of the chip for making low-skew connections over wide areas of the chip.

Then there is the general wiring matrix than consists of short segments of "wires" that interconnect via a small switching matrix. There is a matrix for every CLB. Note that with a fixed amount of routing resources, it is possible to create a design that can't be routed. However, in practice, this is difficult to do. And should that situation arise, a larger chip will solve the problem by permitting a less dense design.

There are also I/O blocks. These provide the actual interface between the IC and the outside world. These are programmable, too, and are not as basic as one might think. Figure 2 shows a simplified drawing of a 3000 series I/O block. As you can see, it's pretty complex for an I/O pin. Both the input and output signals can be registered or direct. The polarity of the output and three-state control line can

FIGURE 1. A Configurable Logic Block (CLB) from a Xilinx 3000 series FPGA. Note that it is similar to a PAL with memory logic instead of an Invert/AND/OR matrix.



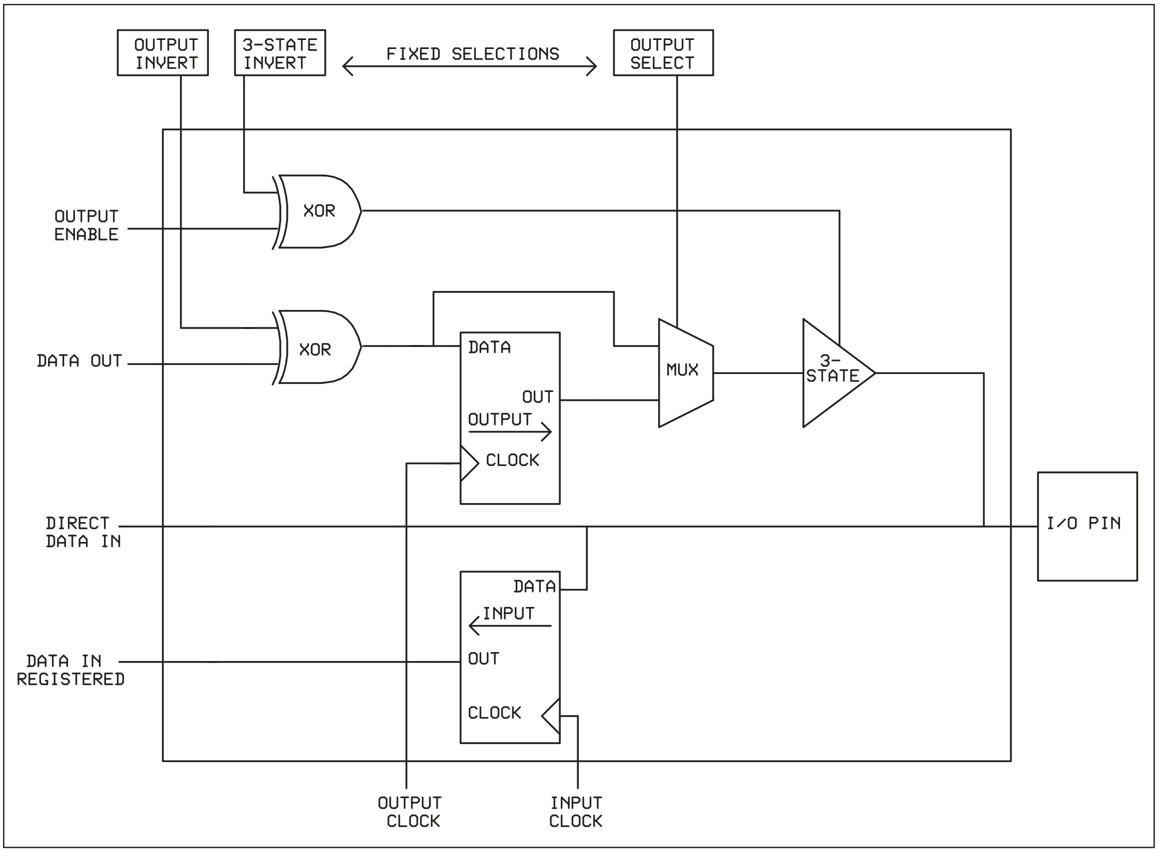


FIGURE 2. An I/O block from a Xilinx 3000 series FPGA. The user has many more options and capabilities when compared to a PAL or CPLD.

be inverted. And there are separate clocks for the input and output. The "Fixed Selections" are those that are set at program time and cannot be changed during normal operation.

But wait! There's more! There are additional Xilinx features available on this and other devices (but not all chips have all the features). This includes internal tri-state busses, the ability to use logic memory as real memory, actual bulk RAM on board, output slew rate control, input switching-level control, and more. If you can do it with discrete ICs, chances are that you can do it better with an FPGA.

Again, different manufacturers have different approaches and implementations. But they all must address the I/O, logic, and interconnect issues in some manner. And they all have additional features that they think are useful and/or important for the design of digital circuits.

Good News — Bad News

It is useful to note that the Xilinx 3000 series is a "mature" technology. It's been around for at least 15 years. The 4000 series and the other Xilinx FPGAs pack two 3000-type CLBs into a single CLB. The smallest 3000 series has 64 CLBs and will replace about 20 standard digital ICs (including counters and registers). Xilinx suggests that this is about equal complexity to the 2000-gate design. Cost for this IC is about $13.

Newer families — like the Virtex and Spartan series — have gate-equivalent counts of 300,000 and more with thousands of double-sized CLBs. This is about 150 times larger and can replace about 3,000 standard digital ICs. That's a whole system on a chip! The unit cost for the Spartan-IIE X2CS300E is under $40. (It now becomes abundantly clear why designing with standard digital ICs is no longer being employed by manufacturers.)

Unfortunately, designing with FPGAs requires software. Naturally, this software is proprietary and can be quite expensive (several thousand dollars or more). That may not be a problem for a manufacturer who can recover that expense from the savings of a single design, but it's not a practical price for most hobbyists. The good news is that these manufacturers realize that it's important to provide entry-level systems so that students and small companies can gain familiarity with their products at low cost. They are in the business of selling ICs — not software.

Cheap or free software systems are available that will allow the user to program at least some of the smaller parts. Typically, these introductory systems do not provide all the advanced features of the full-priced software, but they are certainly adequate for the hobbyist and student. (Note that this software is almost always available as a download from a website or on a CD. Typically, the software consists of hundreds of megabytes of files. So, if you have a slow modem, opt for the CD.)

Resources

FPGA Manufacturers
(not a complete list)

Actel Corporation
www.actel.com

Altera Corporation
The MAX II development board for CPLDs includes a download cable that supports their FPGAs. Cost is about $150.
www.altera.com

Atmel Corporation
www.atmel.com

Lattice Semiconductor Corporation
www.latticesemi.com

QuickLogic Corporation
www.quicklogic.com

Xilinx Corporation
www.Xilinx.com

In addition to the software, some sort of programming hardware is also needed. Again, there are the professional (and expensive) devices and the cheap introductory accessories. Often times this is called a "Download Cable." Download cables are usually about $100. However, before you buy anything, it is very useful to talk to a sales representative. You can usually find them by referencing the manufacturer's website.

Find out exactly what you need. Many times there are evaluation boards or kits that provide all the basic requirements (software and download cable) and are also extremely useful in providing a learning platform. Sometimes the cost of these evaluation boards is equal to (or at least close to) the download cable alone.

Software

There are several methods of getting your design into the system. The traditional way is with a schematic. This is generally supported with a variety of file formats. Most FPGA software systems include a schematic editor with a library of parts. This is very convenient and useful. However, surprisingly, sometimes the library parts are not optimized for the FPGA. This can result in wasted resources which ultimately leads to slower and more costly designs.

The other method is with the use of VHDL, Verilog, ABEL, or one of the other hardware programming languages. Again, an editor is usually provided for at least VHDL. (If not, downloads from the Web can usually be found.)

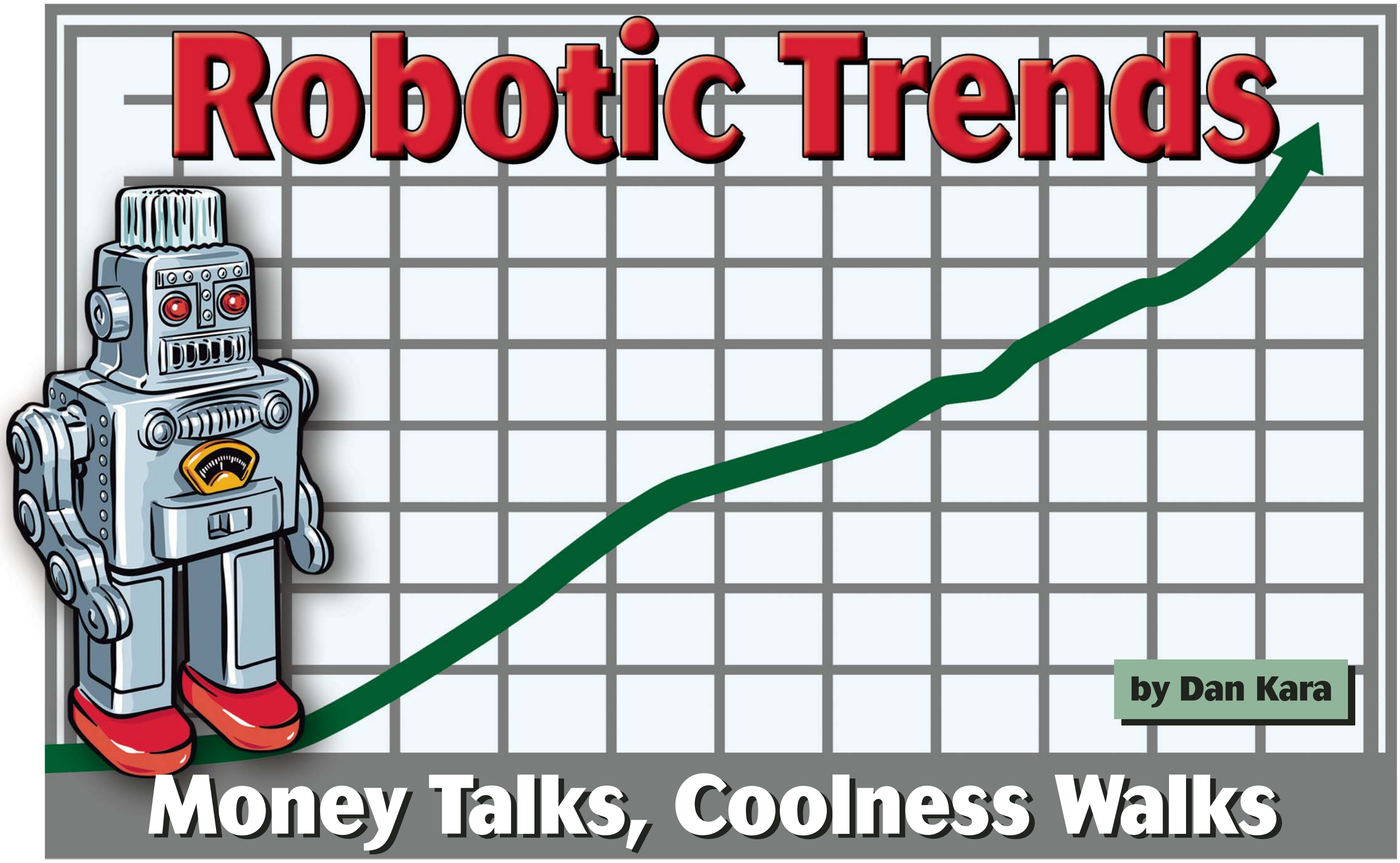
It should be noted that using VHDL or other language for hardware design is not as simple as it may initially appear. Subtle variations can creep in if the designer is not on-guard (a transparent latch instead of a D flip-flop, for example). Such variations can be extremely difficult to identify later.

Additionally, the software approach doesn't define the hardware until compilation time. Sometimes the designer may not have a clue to the real size of the design until it's completed. It may be too large to be practical.

Lastly, ordinary computer software is linear. One thing happens, then another thing, etc. Hardware operates in parallel with many operations occurring simultaneously. VHDL looks like computer software and it's easy to forget that it isn't. You can't debug VHDL with a linear frame of mind. Note that VHDL is a very useful and powerful tool. But it is different from the way most hardware designers work. It's important to understand these differences from the start.

Conclusion

FPGAs are the pinnacle of programmable logic. They provide a flexibility, complexity, and cost effectiveness that cannot be matched with any other off-the-shelf component. There are many manufacturers and many approaches to both the hardware and design software. Introductory systems are usually available at a modest cost that are suitable for many lower-level designs. It is useful for any hobbyist or engineer to understand the capabilities and characteristics of these devices. **SV**



Robotic Trends

Money Talks, Coolness Walks

by Dan Kara

In the robotics industry, "cool" and "career" are not mutually exclusive. However, before you make your robotics career decision, be sure you are placing your bet on markets and companies that know how to make or attract money.

Last month, we explored the subject of a career in robotics. The point was made that it makes little sense to discuss a robotics career without first defining or categorizing the robotics market. To that end, I broke the robotics industry into two major groups — Industrial Robotics and Service Robotics. Service Robotics was further subdivided into Professional Service Robotics, exemplified by technology such as surgical robots and military robots, and Domestic Service Robotics, those robots that find use in the home. Classes of domestic service robots, or if you prefer personal robots/consumer robots, include education/hobbyist robots, home care/lawn care robots, entertainment robots, smart toys, and home assistance /assistive robots.

While defining the market is a necessary first step in any career decision process, it is only a start. You can also apply a taxonomic framework to buggy whips, but I would not want to base a career on that market. It is also necessary to determine if the market is "real." Market "reality" can be determined in any number of ways. One of the best is quantitative market research data. Unfortunately, while quantitative data for the industrial robotics market is numerous and robust, the same cannot be said for the service robotics market. Research is particularly weak in the area of consumer robotics.

Money Talks

How can the robotics industry outside of industrial robotics be validated given the lack of market sizing figures? There is one reality check that trumps all others. How can I put this without being blunt? Actually, there is no way so here goes ... it is all about money. There, I said it. Money. No matter how cool robots and robotic technology is, no matter how many robotics pieces you have seen on the Discovery Channel, and no matter how many competitions and science fair projects you might have participated in, when it comes to careers and career decisions, it is all about the money.

Don't get me wrong. I am not advising that someone pursue or bypass a career based solely on the amount of money they can make. That approach is unwise and usually unfulfilling. There is a reason why the old adage "do what you love and the money will follow" continues to hold true for each new generation coming into the job market.

> **Remember, robots will come into common usage if they can perform a function or service that cannot be performed by humans, or if they can do it more effectively or more cheaply than humans."**

No, when I speak of money I am concerned with an industry/company making money or securing investment dollars. Before anyone agrees to work for a robotics company they should ask themselves "does this company offer products and services that deliver real value to their customers, makes their customers money, or saves them money?" If you cannot answer "yes" to any of these questions, it would serve you better to continue your search for a robotics employer.

Reading Between the Lines

The good news is that the robotics industry is generating investment, products, and revenue. That is, the money is there. Judge for yourself. Following are just a couple of the announcements that came across my desk in a four week period in May-June. Most are straightforward, but hidden within each is the larger message as to where the hot markets in robotics are and why these companies will continue to prosper.

On May 18th, VideoRay of Phoenixville, PA — a maker of Micro Underwater Remotely Operated Vehicles (Micro ROVs) — announced that their VideoRay Pro III product has been delivered to Monterey Bay Aquarium. The announcement — which described how the robot would be used to clean tanks and capture fish — was fairly vanilla. Hidden in the announcement was the fact that the Micro ROV would save the expense of mobilizing divers for many common aquarium procedures.

Remember, robots will come into common usage if they can perform a function or service that cannot be performed by humans, or if they can do it more effectively or more cheaply than humans. While the VideoRay Micro ROV might not perform any better than a diver, their lower cost of operation makes the robot a good business choice.

Companies that have diversified product lines protect themselves from downturns in specific markets. iRobot (Burlington, MA) illustrates this perfectly. The company recently announced that they have sold over two million of the little robot vacuums through over 7,000 retail stores, including Target, Linens 'n Things, Best Buy, Sears, Amazon.com, and Bed, Bath & Beyond, since they were introduced in 2002. Roombas retail for $150 to $330 each, providing iRobot with plenty of revenue and a boatload of validation for the consumer robotics market.

iRobot's military sales have also been strong. For example, a week after iRobot announced its two-millionth Roomba sale, it followed up by disclosing that they had been awarded a $64.3 million Indefinite Delivery-Indefinite Quantity (IDIQ) contract for iRobot PackBot EOD robots, spare parts, training, and repair services.

The award was granted by the Naval Air Warfare Center Training Systems Division (NAVAIR) on behalf of the Robotic Systems Joint Project Office. The robots will be used to support US forces in Iraq, Afghanistan, and elsewhere, to identify and dispose of Improvised Explosive Devices (IEDs).

Also in May, Foster-Miller (Waltham, MA) — iRobot's competitor in the small ground-based military robot market — announced that it has been awarded a $63.9 million IDIQ contract also from NAVAIR for their TALON robots, training, parts, and so on. This contract came on the heels of a contract awarded to Foster-Miller three weeks earlier for an additional $28 million that is part of a separate $257 million NAVSEA IDIQ contract for TALON EOD robots.

Defense spending for robotics technology is not limited to ground vehicles. In fact, in the same May timeframe as the Foster-Miller contract win, Honeywell Defense & Space Electronic Systems received a contract for $61 million from the US Army's Future Combat System (FCS) program lead systems integrators Boeing and partner Science Applications International Corporation (SAIC).

The contract was for the development of Class I Unmanned Aerial Vehicle Systems (UAVS). Class I UAVS — sometimes called micro UAVs — are the smallest of the four unmanned aerial vehicle classes in the FCS program (prototypes weigh in at about 35 pounds). They are designed to hover in

The Global Hawk Unmanned Aerial Vehicle used by the military.

the air providing reconnaissance and surveillance capabilities to soldiers on the ground.

The unmanned aerial vehicles market — of which micro UAVs is only a tiny component — is perhaps the largest robotics market of all. We have all seen full-size UAVs — such as the General Atomics Aeronautical Systems' Predator and Northrop Grumman's Global Hawk at work in Afghanistan and Iraq. These are big budget items in the robotics industry, but relatively cheap in terms of military aircraft. Therein is their appeal. Not only do UAVs exhibit an extremely high mission effectiveness rating, they are much cheaper to purchase, fly, and maintain that manned aircraft.

Corrections in the Military Robotics Market?

While the military robotics market is going gangbusters now, there is always the possibility that there might be a downturn. Some analysts are already projecting that the defense industry will go into a correction when funding and sales begin to decline in the latter part of this decade. The logic goes that programs whose development and production were accelerated to deal with the war on terrorism in Afghanistan and Iraq will decline after they have reached completion.

A dramatic drop in spending on military robotics is a real possibility, and might impact the robotics industry as a whole. However, robots — unlike other types of military hardware — can have many civilian applications. For example, while unmanned aerial vehicles are nearly exclusively used for military purposes, non-military uses of UAVs are being explored by many companies for use in a vast number of civilian applications.

Cool Does Not Count for Much

It should be noted that market downturns are not limited to the defense industry. Any industry — with the possible exception of the porn and healthcare industries — are subject to the vicissitudes of capitalism's invisible hand. The important things to look for are robotics companies and markets that are attracting investment or generating revenue. Both sources of money are based on business and product plans that deliver value to customers, make them money, or save them money. In the real world of careers, kids, and mortgages, the fact that robots are cool counts for little. The great thing about the robotics industry, however, is that robots and robotics technologies are cool and solid business can be built around them. **SV**

Dan Kara is President of Robotics Trends, the producer of the RoboBusiness (**www.robobusiness2006.com**) and RoboNexus (**www.robonexus.com**) conferences, and publisher of Robotics Trends (**www.roboticstrends.com**), an online news, information, and analysis portal covering the personal, service, and mobile robotics market. He can be reached at dk@roboticstrends.com

It's Your Bag

Have you ever been looking for a specific tool in your workshop — like a screwdriver — only to find a hammer, instead? Then you did the unthinkable — you hammered a screw into something. Ugh. What you need is a better tool storage system. What you really should get is an eight-inch Electrician's/Maintenance Tote (Model #22128) from McGuire-Nicholas®. A great transportable tool storage solution.

McGuire-Nicholas is a product label of the Rooster® Group. A well-known supplier of plastic storage systems, suspenders, and back support belts for carpenters, electricians, plumbers, and other Do-It-Yourself (DIY) workers, you can purchase McGuire-Nicholas products at your favorite home improvement supplier. For more information, go to **www.mcguire-nicholas.com**

Built from a rugged wear material — that McGuire-Nicholas calls Toughwear™ — this tote sports over 30 pockets, loops, hooks, and latches for holding just about every robot-building tool that you own. Additionally, an included plastic parts organizer can be used for holding those small parts and fasteners that typically slide around and hide inside other tool storage bags.

The McGuire-Nicholas Electrician's/Maintenance Tote.

One feature that we really like about this tote is that it stands upright when placed flat on the ground. Then just push aside the oversized carrying handle and you have easy access to the four big pockets that are ensconced inside the tote's main body.

If you're looking for a storage solution that is a little less vertical and more horizontal, the 14-inch Tool Bag with Plastic Tray (Model #22314) offers fewer loops, hooks, and latches, but more "super-sized" pockets for holding your batteries, wheels, and servo motors. All wrapped up in an easy to transport bag.

My Other Parts Storage is a Porsche (Red)

No matter what your level of involvement is with robotics — BEAM, combat, microcontroller, or LEGO® Mindstorms® — one thing is for sure, after about one year of building bots, you will have acquired a substantial warehouse of parts, components, and elements. Now your real problem arises ... where to store all of this stuff.

It's not just a matter of ferreting your parts collection away in a set of plastic cubbyholes, you have to be able to find the right part at the right time. What you need is a parts organizer storage system.

As the leader in the manufacture of tackle boxes, Plano Molding Company (**www.planomolding.com**) has over 50 years' worth of experience in designing practical and useful plastic storage systems. This experience is readily apparent in their incredible Stow 'N Go™ product line.

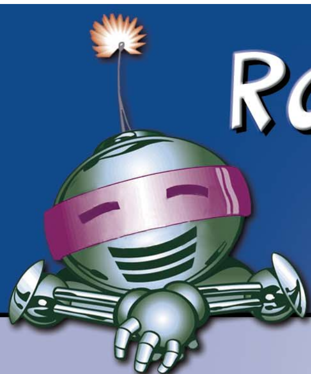
Starting with the single-sided Stow 'N Go Organizer (5230), you will be able to segregate up to 27 different parts. Protected with a smooth-operating snap closure system that Plano Molding calls Lock-Jaw™, this organizer has a clear hinged lid for quick "see-through" parts identification. Painted in a fancy color called Porsche Red, this impact-resistant organizer will make you feel like your components collection is being transported in a Hummer.

If you were raised on a tackle box mentality, however, the Stow 'N Go Pro Rack Organizer (1354-20) will give you an instant familiarity check. Looking a lot like a tall tackle box, the Pro Rack Organizer is actually four Plano Molding 3500 series StowAway® utility boxes housed inside a brilliant smoked plastic drop-front lid. Just flip two safety latches, drop the front lid, and slide out one of the four utility boxes. Access to your beloved LEGO Technic elements has never been so easy.

The Plano Molding Company Stow 'N Go.

Additionally, there is an open bay storage bin located in the Pro Rack Organizer's lid. An oversized snap closure latch keeps this lid shut during transport.

If you're one of *SERVO Magazine*'s founding subscribers, then you have probably amassed a much larger parts collection. Luckily, Plano Molding makes a couple of larger storage solutions (e.g., Stow 'N Go Double Organizer; 5232 and Stow 'N Go Pro Rack Organizer w/3600 boxes; 1364-20) — just right for your "big boy" toys. **SV**



ROBOTICS RESOURCES

Tune in each month for a heads-up on where to get all of your "robotics resources" for the best prices!

BY GORDON MCCOMB

Robotic Arms and Grippers

Imagine going through life with your arms in a straightjacket. Now imagine how your armless, handless robot feels — er, assuming it has feelings to begin with! Robots without appendages are limited to rolling or walking about, possibly noting things than occur around them, and little else.

The more sophisticated robots in science, industry, and research have at least one arm for the purpose of grasping, moving, or reorienting objects. Arms extend the reach of robots and make them more human-like. For all the extra capabilities they provide a robot, it's notable that arms and hands aren't too difficult to add. You can build your own, or purchase readymade solutions from a number of sources.

This installment of Robotics Resources deals with the concept and design theory of basic robotic arms and grippers. We'll stay on the low end of the cost spectrum, suitable for use on smaller desktop and educational 'bots. We'll also consider as separate entities the arm and hand (or "gripper") mechanisms. Each provide their own distinct functions.

A Look at Human Arms

Consider for a moment your own arms. You'll probably notice a number of important points. First, your arm has two major joints: the shoulder and the elbow (the wrist, as far as robotics is concerned, is usually considered part of the gripper mechanism). Your shoulder can move in two planes — both up and down, and back and forth. The elbow joint is capable of moving in two planes, as well: back and forth, and up and down.

The joints in your arm, and your ability to move them, are called *degrees of freedom*. Your shoulder provides two degrees of freedom in itself: shoulder rotation and shoulder flexion. The elbow joint adds a third and fourth degree of freedom: elbow flexion and elbow rotation.

Robotic arms also have degrees of freedom. But instead of muscles, tendons, ball-and-socket joints, and bones, robot arms are made from metal, plastic, wood, motors, solenoids, gears, pulleys, and a variety of other mechanical components. Some robot arms provide just one degree of freedom. Others provide three, four, and even more separate degrees of freedom.

Types of Robotic Arms

In the world of robotics, arms are classified by the shape of the area that the end of the arm (where the gripper is) can reach. This accessible area is called the *work envelope*. For the most part, in the case of a mobile robot, the work envelope does not take into consideration motion by the robot's body, just the arm mechanism.

The human arm has a nearly spherical work envelope. We can reach just about anything, as long as it is within arm's length, within the inside of about three-quarters of a sphere. In a robot, such a robot arm would be said to have revolute coordinates. There are three more, decidedly non-human robot arm designs: polar coordinate, cylindrical coordinate, and Cartesian coordinate. All of them support at least three degrees of freedom.

As noted above, revolute coordinate robotic arms are modeled after the human arm, so they have many of the same capabilities. The typical design is somewhat different, however, because of the complexity of the human shoulder joint.

In the *revolute coordinate* arm, the shoulder joint of the robotic arm is really two different mechanisms. Shoulder rotation is accomplished by spinning the arm at its base, almost as if the arm were mounted on a turntable. Shoulder flexion is accomplished by tilting the upper arm member backward and forward. Elbow flexion works just as it does in the human arm. It moves the forearm up and down. Revolute coordinate arms are a favorite design choice for hobby robots. They provide a great deal of flexibility, and provide an appearance similar to that of the human arm.

The work envelope of the *polar coordinate* arm is half-sphere shaped. Polar coordinate arms are among the most flexible in terms of being able to grasp a variety of objects scattered about the robot. A turntable rotates the entire arm, just as it does with a revolute coordinate arm. This function is akin to shoulder rotation. The polar coordinate arm lacks a means for flexing or bending its shoulder, however.

The second degree of freedom is the elbow joint, which moves the forearm up and down. The third degree of freedom is accomplished by varying the reach of the forearm. An "inner" forearm extends or retracts to bring the gripper closer to or farther away from the robot. Without the inner forearm, the arm would be able to grasp objects laid out in a finite two-dimensional circle in front of it.

The *cylindrical coordinate* arm looks a little like a robotic forklift. The name is derived from the cylindrical shape of its work envelope. Shoulder rotation is accomplished by a revolving base, as in revolute and polar coordinate arms. The forearm is attached to an elevator-like lift mechanism. The forearm moves up and down this column to grasp objects at various heights. To allow the arm to reach objects in three-dimensional space, the forearm is outfitted with an extension mechanism, similar to the one found in a polar coordinate arm.

The work envelope of a *Cartesian coordinate* arm resembles a box. It is the most unlike the human arm and least resembles the other three arm types. It has no rotating parts. The base consists of a conveyer belt-like track. The track moves the elevator column (like the one in a cylindrical coordinate arm) back and forth. The forearm moves up and down the column and has an inner arm that extends the reach closer to, or farther away from, the robot.

Powering the Arm

There are three general ways of moving the joints in a robot arm:

• *Electrical actuation* is with the use of motors, solenoids, and other electro-mechanical devices. It is the most common and easiest to implement. The motors for elbow flexion, as well as the motors for the gripper mechanism, can be placed in or near the base. Cables, chains, or belts connect the motors to the joints they serve.

• *Hydraulic actuation* uses oil-reservoir pressure cylinders, similar to the kind used in earth-moving equipment and automobile brake systems. The fluid is non-corrosive and inhibits rust: both are the immediate ruin of any hydraulic system. Though water can be used in a hydraulic system, if the parts are made of metal, no doubt they will eventually suffer from rust, corrosion, or damage by water deposits. For a simple home-brew robot, however, a water-based hydraulic system using plastic parts is a viable alternative.

• *Pneumatic actuation* is similar to hydraulic, except that pressurized air is used instead of oil or fluid (the air often has a small amount of oil mixed in it for lubrication purposes). Both hydraulic and pneumatic systems provide greater power than electrical actuation, but they are more difficult to use. In addition to the actuation cylinders themselves, a pump is required to pressurize the air or oil, and valves are used to control the retraction or extension of the cylinders.

Note that electrical activation doesn't always have to be via an electro-mechanical device such as a motor or solenoid. Other electrically-induced activation is possible using a variety of technologies. One of particular interest to hobby robot builders is shape memory alloy, or SMA. SMA material goes by a number of trade names, such as Bio-Metal, Dynalloy, Nitinol, and Muscle Wire.

The construction of various SMA materials differs from manufacturer to manufacturer, but the activation technique is about the same: when heat is apply to the metal, it contracts to a pre-defined state. Heat can be applied directly, through a flame or with hot water, or by passing an electrical current through the material. Electrical activation is the most common technique used in robotics.

A disadvantage of SMA is its slow expansion rate: the metal must cool before it relaxes and returns to its pre-heated shape and size. The larger the metal, the longer it takes to cool, so the slower the "muscle" returns to its non-contracted state. As a result, most of the shape memory alloy material you'll see available is hair-thin. Don't let the small diameter of the wire fool you, however. Muscle Wire, and many other SMA materials, can hold considerable weight — several pounds in both the contracted and non-contracted state.

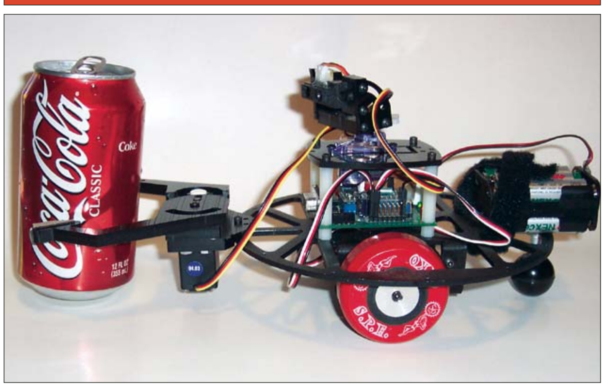
An interesting variation on pneumatic actuation is the "Air Muscle," an ingenious combination of a small rubber tube and black plastic mesh. The rubber tube acts as an expandable bladder, and the plastic mesh forces the tube to inflate in a controllable manner.

Air Muscle is available pre-made in various sizes. It is activated by pumping air into the tube. When filled with air, the tube expands its width, but contracts its length by about 25 percent. The result is that the tube and mesh act as a kind of mechanical muscle. The Air Muscle is said to be more efficient than the standard pneumatic cylinder and, according to its makers, has about a 400:1 power-to-weight ratio.

A Closer Look at Robotic Grippers

In the world of robotics, hands are usually called *grippers*, or *end effectors.* The word gripper better describes their function. Few robotic hands can manipulate objects with the fine motor control of a human hand. They simply grasp or grip the object, hence the name gripper.

FIGURE 1. Robodyssey's Gripper mounted onto a mouse and holding a can with eight ounces of water.



FOR YOUR INFO

In addition to the companies listed in this article, the following online sources resell arm and gripper kits made by OWI and/or other manufacturers. A few offer their own custom products:

www.hobbyengineering.com
www.imagesco.com
www.robotstore.com

Gripper designs are numerous, and none are ideal for all applications. Each gripper style has unique advantages. Here are a number of useful gripper designs you can use for your various robots, but this list is by no means inclusive of all the gripper styles. As a quick note, the names I use to refer to the various gripper designs are descriptive only. You'll see a variety of terms used for these and other forms of grippers.

Snap-Activated Lever (the "Clapper")

The snap-activated lever is one of the simplest gripper designs. The gripper is composed of two metal, wood, or plastic plates. The bottom plate is secured to the gripper body; the top plate is hinged. A small spring-loaded solenoid is positioned inside, between the two plates. When the solenoid is not activated, a spring pushes the two flaps out, and the gripper is open. When the solenoid is activated, the plunger pulls in, and the gripper closes. The amount of movement at the end of the gripper is minimal — about a half inch with most solenoids. However, that is enough for general gripping tasks.

Variations include using a small motor instead of the solenoid. A solenoid provides for either open or closed positions, whereas a proper motor allows for positions in between. For example, by using a small R/C servo motor, it's possible to position the movable plate to various points. The movable plate is connected to the servo motor by way of a linkage.

Pincher

The pincher gripper uses two movable "fingers" that open or close, either from a common hinging point, or with a special joint that allows for parallel movement of the fingers. The benefit of the pincher is that both fingers close in around an object. This is generally preferred over the snap-activated lever, where one plate is fixed and the other moves.

The designs of pinchers can vary greatly. One common approach is to use a pull-rod at the base of the fingers. Pull on the pull-rod, and the fingers open; push on it, and they close. The action of the fingers is like the blades of a pair of scissors, which means for larger and rounded objects, there is a potential that the closing fingers will actually push the object away. This problem can be largely avoided by using fingers shaped in a semi-circle so that the fingers close in around the object more evenly.

With a parallel joint, the fingers are parallel from one another throughout their entire in-and-out movement. A disadvantage of most parallel pincher grippers is that they use straight fingers, which afford little contact area when grasping round and cylindrical objects.

Flexible Finger

Various designs are used to produce a gripper with human finger-like appendages. The fingers are usually just segmented pieces of plastic, wood, or metal, with a thin flap for a hinge. A very thin strip of metal (sometimes plastic) is placed on the inside of the fingers. When the strip is pulled taut, the fingers close in. The strip is naturally springy, so when it's released, the fingers open back up. Several child toys have used this design principle, and in fact, such toys are often re-engineered for use on a homemade robot.

FIGURE 2. Joinmax/MCIIRobot sells plastic robot kits, including an arm/gripper with six degrees of freedom. The arm is controlled by specialized servo motors.

Adding Wrist Rotation

The human wrist has three degrees of freedom: it can twist on the forearm, it can rock up and down, and it can rack from side to side. You can add some or all of these degrees of freedom to a robotic hand. With most arm designs, you'll just want to rotate the gripper at the wrist.

Wrist rotation is usually performed by a motor attached at the end of the arm or at the base. When connected at the base (for weight considerations), a cable or chain joins the motor shaft to

the wrist. The gripper and motor shaft are outfitted with mating spur gears. You can also use chains (miniature or #25) or timing belts to link the gripper to the drive motor.

Another possibility is a worm gear on the motor shaft. Remember that worm gears introduce a great deal of gear reduction, so take this into account when planning your robot. The wrist should not turn too quickly or too slowly.

Yet another approach is to use a rotary solenoid. These special-purpose solenoids have a plate that turns 30–50 degrees in one direction when power is applied. The plate is spring loaded, so it returns to normal position when the power is removed. Mount the solenoid on the arm and attach the plate to the wrist of the gripper.

Sources

For those who wish to purchase a ready-made arm and/or gripper, there are a number of commercial enterprises that offer low-cost solutions. The following list is by no means exhaustive, but it should point you in the right direction.

Budget Robotics
www.budgetrobotics.com

My own small robotics manufacturing company, Budget Robotics offers several very low-cost grippers. Products include the Big Gripper, a scissor-style pincher gripper with formed fingers.

CrustCrawler
www.crustcrawler.com

Five- and six-axis robotic arms, operated by R/C servos. Crustcrawler kits are made from aluminum.

Joinmax/MCIIRobot
www.mciirobot.com

US distributor of Joinmax kits — includes a robot arm and grip kit with six degrees of freedom. Resellers include **Pololu.com**, **Garage-Technologies.com**, and many others.

Lynxmotion
www.lynxmotion.com

Manufactures and sells numerous servo-based arm and gripper kits.

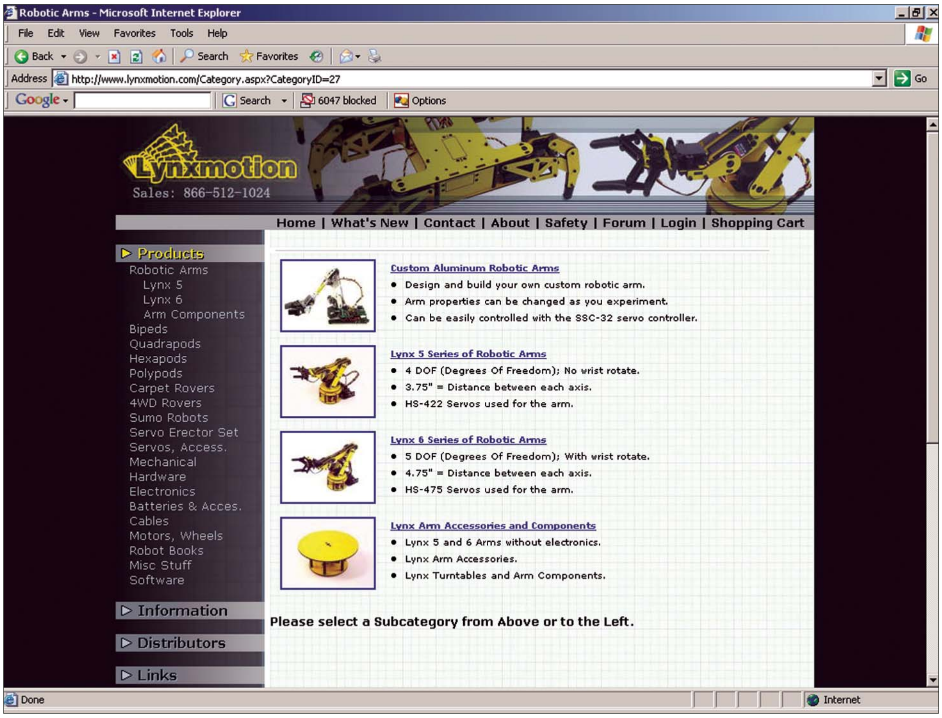


FIGURE 3. Lynxmotion arm kits are made from sturdy polycarbonate. The company also offers brackets and hardware to make custom-designed, servo-operated arms.

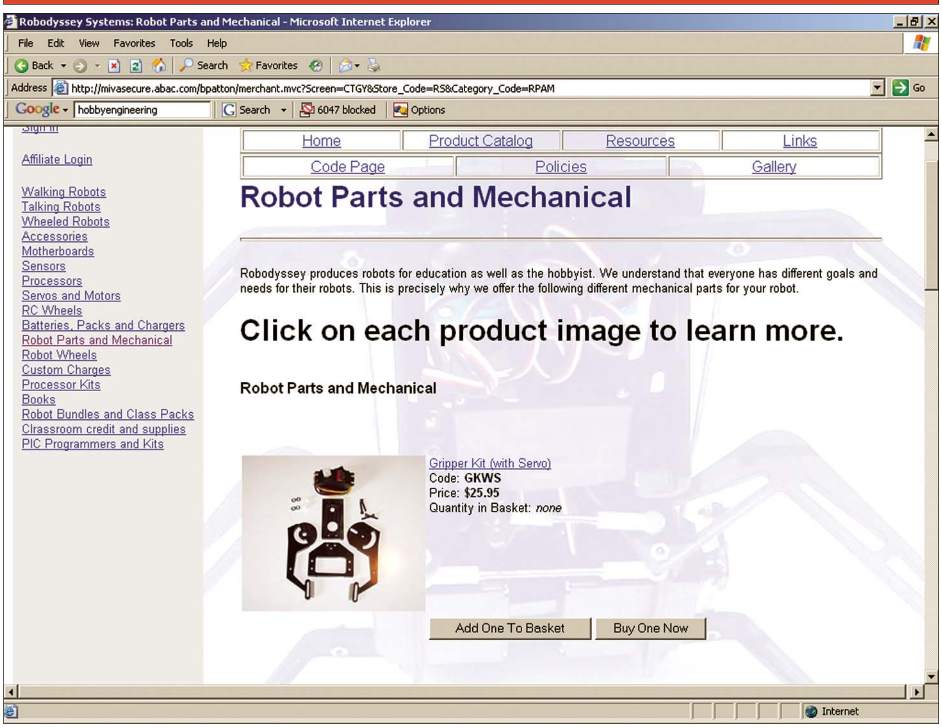
Parallax, Inc.
www.parallax.com

Boe-Bot Gripper Kit, five-axis robotic arm (resold from Crustcrawler).

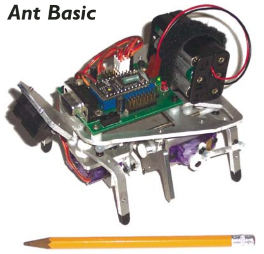
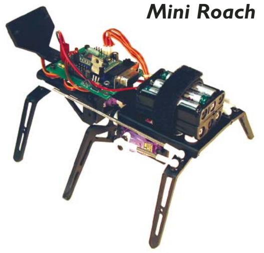
Robodyssey
www.robodyssey.com

Offers a low-cost, scissor-style, servo-operated pincher gripper. **SV**

FIGURE 4. Robodyssey's gripper kit includes a servo and shaped parts to make a scissor-style pincher.



THREE SERVO HEXAPOD ROBOT KITS

MANUFACTURER	Robot Name	Model Number	Overall Length	Overall Width	Overall Height	Ground Clearance	Servos	Servo Torque	Weight (w/o batteries)	Deck Width for Electronics	Deck Length for Electronics	Body Material	Leg Material	# of Standard Sensors
Images SI, Inc. www.imagesco.com	Hexapod-01	Hex-01	11	7	4	2.25	Hitec HS-325HB	51 oz-in	12	2.5	3	Aluminum and Acrylic	Acrylic	2
 *Extreme Hexapod I – Basic ATOM*	Extreme Hexapod I – Basic ATOM	EH1AC-KT	9.5	11	4	1.75	Hitec HS-422	57 oz-in	17.6 oz	3	5	Lexan	Lexan	1
Lynxmotion www.lynxmotion.com	Extreme Hexapod I – BASIC Stamp	EH1BC-KT	9.5	11	4	1.75	Hitec HS-422	57 oz-in	17.6 oz	3	5	Lexan	Lexan	1
	Extreme Hexapod I – Bare Bones	EH1-KT	9.5	11	4	1.75	Hitec HS-422	57 oz-in	17.6 oz	3	5	Lexan	Lexan	1
Ant Basic 	Ant Basic	ANTB	4.75	4	3.38	0.13	Blue Bird BMS 380	50 oz-in	8.5	3.88	3.88	Aluminum	Aluminum	1
	Ant-n-PICPac Combo	ANTPP	4.75	4	3.38	0.13	Blue Bird BMS 380	50 oz-in	8.5	3.88	3.88	Aluminum	Aluminum	1
	Ant – Bare Bones	ANTBB	4.75	4	3.38	0.13	Blue Bird BMS 380	50 oz-in	8.5	3.88	3.88	Aluminum	Aluminum	1
Robodyssey www.robodyssey.com	Basic Roach	BR	9.75	8.5	4	0.88	Futaba S3004	44.4 oz-in	14	2.5	8.25	Aluminum	Aluminum	1
Mini Roach 	Basic Roach – Experimenters	RE	9.75	8.5	4	0.88	Futaba S3004	44.4 oz-in	14	2.5	8.25	Aluminum	Aluminum	1
	Basic Roach – Bare Bones	BBR	9.75	8.5	4	0.88	Futaba S3004	44.4 oz-in	14	2.5	8.25	Aluminum	Aluminum	1
	Mini-Roach	RMB	7.63	6.75	3.38	0.75	Blue Bird BMS 380	50 oz-in	8	2	6.13	Aluminum	Aluminum	1
	Mini-Roach – Bare Bones	RMBB	7.63	6.75	3.38	0.75	Blue Bird BMS 380	50 oz-in	8	2	6.13	Aluminum	Aluminum	1
Yost Engineering, Inc. www.yostengineering.com	BugBrain Walking Robot	BBBW	15	10	5	2	Tower TS-53	42 oz-in	9	3	15	Fiberglass	Aluminum	2

MATRIX

by Pete Miles

All product photos are available on the *SERVO* website at www.servomagazine.com

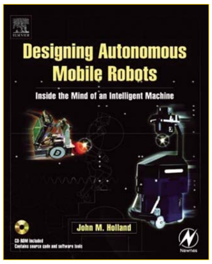
Upcoming topics include SBCs and H-bridges, sensors, kits, and actuators. If you're a manufacturer of one of these items, please send your product information to: **BrainMatrix@servomagazine.com** Disclaimer: Pete Miles and the publishers strive to present the most accurate data possible in this comparison chart. Neither is responsible for errors or omissions. In the spirit of this information reference, we encourage readers to check with manufacturers for the latest product specs and pricing before proceeding with a design. In addition, readers should not interpret the printing order as any form of preference; products may be listed randomly or alphabetically by either company or product name.

Sensor Type	Sensor Model #	Standard Microcontroller	Microcontroller Manufacturer	Memory	Digital I/O Pins	Analog I/O Pins	Programming Language(s)	Preprogrammed Motion Routines	Programming Interface	Base Robot Kit Price (Robot Body & Servos)	List Price of Base Robot With Electronics	Comments
Contact	N/A	PIC 16F628	Microchip	2K	13	0	PICBasic or any other Microchip Programming Language	Yes	None	Not Available	106.95	Includes CdS Light Sensors and programs for light following and avoidance.
Infared Distance	IRPD-01	Basic ATOM	Basic Micro	14K	20	4	Basic	Yes	Serial	—	224.75	Includes the Mini Atom Bot Board which supports most 5V sensors and the Playstation 2 controller remote control, programming cable.
Infared Distance	IRPD-01	BASIC Stamp 2	Parallax	2K-16K	16	0	Pbasic	Yes	Serial	—	213.80	Includes the Mini Atom Bot Board which supports most 5V sensors and the Playstation 2 controller remote control, programming cable.
Infared Distance	IRPD-01	BASIC Stamp 2	Parallax	2K-16K	16	0	Pbasic	Yes	Serial	99.95	—	Precision laser cut parts for durable construction.
Infared Distance	Sharp GP2YOA21	BasicX-24p	NetMedia BasicX	32K	21	8	BasicX	No	Serial	—	199.95	Includes Robodyssey Advanced Motherboard, batteries and battery pack, programming cable.
Infared Distance	Sharp GP2YOA21	PIC 16F876A	Microchip	14K	22	6	PICBasic or any other Microchip Programming Language	No	Serial/ USB	—	149.95	Includes PICPac Motherboard, batteries, and battery pack.
Infared Distance	Sharp GP2YOA21	PIC 16F876A	Microchip	14K	22	6	PICBasic or any other Microchip Programming Language	No	Serial/ USB	139.95	—	Heavy-duty construction that is school tough; all documentation is curriculum based.
Infared Distance	Sharp GP2YOA21	BasicX-24p	NetMedia BasicX	32K	21	8	BasicX	No	Serial	—	299.95	Includes Robodyssey Advanced Motherboard, batteries and battery pack, programming cable.
Infared Distance	Sharp GP2YOA21	BasicX-24p	NetMedia BasicX	32K	21	8	BasicX	No	Serial	—	356.10	Includes Robodyssey Advanced Motherboard, batteries and battery pack, Experimenters Solderless BreadBoard, breadboard cables, standoff, programming cable.
Infared Distance	Sharp GP2YOA21	BasicX-24p	NetMedia BasicX	32K	21	8	BasicX	No	Serial	192.95	—	Heavy-duty construction that is school tough; all documentation is curriculum based.
Infared Distance	Sharp GP2YOA21	BasicX-24p	NetMedia BasicX	32K	21	8	BasicX	No	Serial	—	239.50	Includes Robodyssey Advanced Motherboard, batteries and battery pack, programming cable.
Infared Distance	Sharp GP2YOA21	BasicX-24p	NetMedia BasicX	32K	21	8	BasicX	No	Serial	139.95	—	Heavy-duty construction that is school tough; all documentation is curriculum based.
Contact	PLHS	BasicX-24p	NetMedia BasicX	32K	21	8	BasicX	Yes	Serial	Not Available	179.00	Includes one speaker, three programmable buttons, six LEDs, power supply for testing, compiler on a CD, newsletter on new applications, remote control capable.

Mind Candy For Today's Roboticist

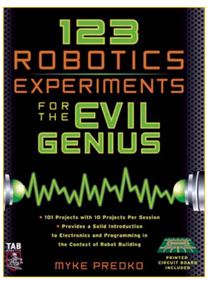
The SERVO Store

Designing Autonomous Mobile Robots
by John Holland



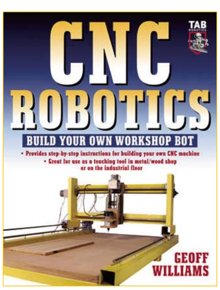
Designing Autonomous Mobile Robots introduces the reader to the fundamental concepts of this complex field. The author addresses all the pertinent topics of the electronic hardware and software of mobile robot design, with particular emphasis on the more difficult problems of control, navigation, and sensor interfacing. Covering topics such as advanced sensor fusion, control systems for a wide array of application sensors and instrumentation, and fuzzy logic applications, this volume is essential reading for engineers undertaking robotics projects, as well as undergraduate and graduate students studying robotic engineering, artificial intelligence, and cognitive science. **$49.95**

123 Robotics Experiments for the Evil Genius
by Myke Predko



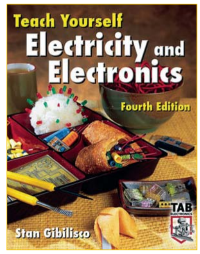
If you enjoy tinkering in your workshop and have a fascination for robotics, you'll have hours of fun working through the 123 experiments found in this innovative project book. More than just an enjoyable way to spend time, these exciting experiments also provide a solid grounding in robotics, electronics, and programming. Each experiment builds on the skills acquired in those before it so you develop a hands-on, nuts-and-bolts understanding of robotics — from the ground up. **$25.00**

CNC Robotics
by Geoff Williams



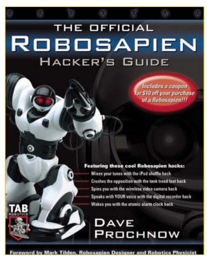
CNC Robotics gives you step-by-step, illustrated directions for designing, constructing, and testing a fully functional CNC robot that saves you 80 percent of the price of an off-the-shelf bot — and that can be customized to suit your purposes exactly, because you designed it. Written by an accomplished workshop bot designer/builder, this book gives you all the information you'll need on CNC robotics! **$34.95**

Teach Yourself Electricity and Electronics — Fourth Edition
by Stan Gibilisco



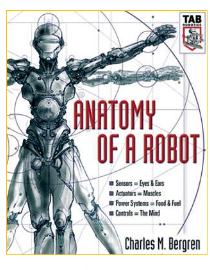
Learn the hows and whys behind basic electricity, electronics, and communications without formal training. The best combination self-teaching guide, home reference, and classroom text on electricity and electronics has been updated to deliver the latest advances. Great for preparing for amateur and commercial licensing exams, this guide has been prized by thousands of students and professionals for its uniquely thorough coverage ranging from DC and AC concepts to semiconductors and integrated circuits. **$34.95**

The Official Robosapien Hacker's Guide
by Dave Prochnow



The Robosapien robot was one of the most popular hobbyist gifts of the 2004 holiday season, selling approximately 1.5 million units at major retail outlets. The brief manual accompanying the robot covered only basic movements and maneuvers — the robot's real power and potential remain undiscovered by most owners — until now! This timely book covers all the possible design additions, programming possibilities, and "hacks" not found anyplace else. **$24.95**

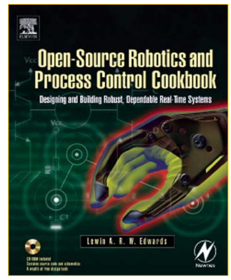
Anatomy of a Robot
by Charles Bergren



This work looks under the hood of all robotic projects, stimulating teachers, students, and hobbyists to learn more about the gamut of areas associated with control systems and robotics. It offers a unique presentation in providing both theory and philosophy in a technical, yet entertaining way. Reading *Anatomy of a Robot* is like having a robot on the operating room table. Crack open the pages and you'll be able to dissect a robot from head to toe. **$29.95**

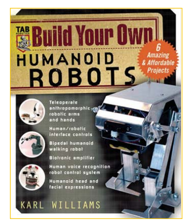
We accept VISA, MC, AMEX, and DISCOVER
Prices do not include shipping and may be subject to change.

Open-Source Robotics and Process Control Cookbook
by Lewin Edwards



In this comprehensive guide, experienced embedded engineer and author, Lewin Edwards, demonstrates efficient and low-cost open-source design techniques, covering end-to-end robotic/process control systems using Linux as the development platform (and also as the embedded operating system), with extensive information on free compilers and other tools. Specifically, the book targets development of real-time physical system controls using Atmel AVR microcontrollers communicating with Linux-based PCs for overmonitoring. Code examples are given to provide concrete illustrations of tasks described in the text. The accompanying CD-ROM contains all the code used in the design examples, as well as useful open-source tools for robotics and process control system design. **$49.95**

Build Your Own Humanoid Robots
by Karl Williams



This unique guide to sophisticated robotics projects brings humanoid robot construction home to the hobbyist. Written by a well-known figure in the robotics community, *Build Your Own Humanoid Robots* provides step-by-step directions for six exciting projects, each costing less than $300. Together, they form the essential ingredients for making your own humanoid robot. **$24.95**

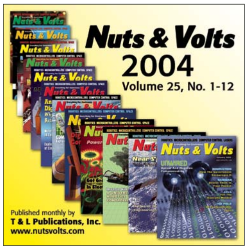
101 Spy Gadgets for the Evil Genius
by Brad Graham/Kathy McGowan

Utilizing inexpensive, easily-obtainable components, you can build the same information gathering, covert sleuthing devices used by your favorite film secret agent. Projects range from simple to sophisticated and come complete with a list of required parts and tools, numerous illustrations, and step-by-step assembly instructions. **$24.95**

To order call 1-800-783-4624 or go to our website at www.servomagazine.com

Mind Candy For Today's Roboticist

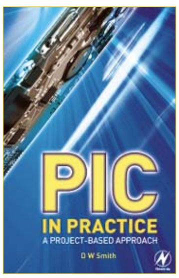
Nuts & Volts CD-Rom



Here's some good news for *Nuts & Volts* readers! Starting with the January 2004 issue of *Nuts & Volts*, all of the issues through the 2004 calendar year are now available on a CD that can be searched, printed, and easily stored. This CD includes all of Volume 25, issues 1-12, for a total of 12 issues. The CD-Rom is PC and Mac compatible. It requires Adobe Acrobat Reader version 6 or above. Adobe Acrobat Reader version 7 is included on the disc. **$29.95**

PIC in Practice: A Project-based Approach — Second Edition

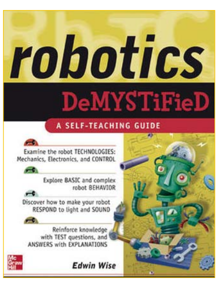
by David W. Smith



PIC in Practice is a graded course based around the practical use of the PIC microcontroller through project work. Principles are introduced gradually, through hands-on experience, enabling students to develop their understanding at their own pace. The book can be used at a variety of levels and the carefully graded projects make it ideal for colleges, schools, and universities. Newcomers to the PIC will find it a painless introduction, while electronics hobbyists will enjoy the practical nature of this first course in microcontrollers. **$29.95**

Robotics Demystified

by Edwin Wise



There's no easier, faster, or more practical way to learn the really tough subjects. McGraw-Hill's Demystified titles are the most efficient, interestingly written, brush-ups you can find. Organized as self-teaching guides, they come complete with key points, background information, questions at the end of each chapter, and even final exams. You'll be able to learn more in less time, evaluate your strengths and weaknesses, and reinforce your knowledge and confidence. **$19.95**

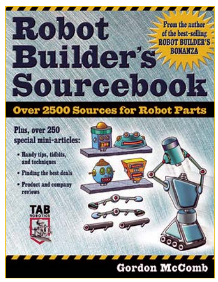
Check out our online bookstore at www.servomagazine.com for a complete listing of all the books that are available.

SERVO CD-Rom

Are you ready for some good news? Starting with the first *SERVO Magazine* issue — November 2003 — all of the issues through the 2004 calendar year are now available on a CD that can be searched, printed, and easily stored. This CD includes all of Volume 1, issues 11-12 and Volume 2, issues 1-12, for a total of 14 issues. The CD-Rom is PC and Mac compatible. It requires Adobe Acrobat Reader version 6 or above. Adobe Acrobat Reader version 7 is included on the disc. **$29.95**

Robot Builder's Sourcebook

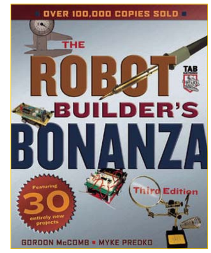
by Gordon McComb



Fascinated by the world of robotics, but don't know how to tap into the incredible amount of information available on the subject? Clueless as to locating specific information on robotics? Want the names, addresses, phone numbers, and websites of companies that can supply the exact part, plan, kit, building material, programming language, operating system, computer system, or publication you've been searching for? Turn to *Robot Builder's Sourcebook* — a unique clearinghouse of information that will open 2,500+ new doors and spark almost as many new ideas. **$24.95**

Robot Builder's Bonanza Third Edition

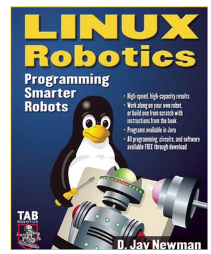
by Gordon McComb / Myke Predko



Everybody's favorite amateur robotics book is bolder and better than ever — and now features the field's "grand master" Myke Predko as the new author! Author duo McComb and Predko bring their expertise to this fully-illustrated robotics "bible" to enhance the already incomparable content on how to build — and have a universe of fun — with robots. Projects vary in complexity so everyone from novices to advanced hobbyists will find something of interest. Among the many new editions, this book features 30 completely new projects! **$27.95**

Linux Robotics

by D. Jay Newman



If you want your robot to have more brains than microcontrollers can deliver — if you want a truly intelligent, high-capability robot — everything you need is right here. *Linux Robotics* gives you step-by-step directions for "Zeppo," a super-smart, single-board-powered robot that can be built by any hobbyist. You also get complete instructions for incorporating Linux single boards into your own unique robotic designs. No programming experience is required. This book includes access to all the downloadable programs you need, plus complete training in doing original programming. **$34.95**

$20 each 15% off if you buy two or more.

From HomoSapien to RoboSapien

Before R2D2 there was R1D1

SERVO Magazine T-Shirts

Get your very own limited-edition *SERVO Magazine* T-shirt. Shirts come in sizes S, M, L, and XL, and are available in either black or white. All shirts are 100% pre shrunk cotton.

MENAGERIE

Technical Titans Face Off in Texas

Corey Morris

Fear Factor gives Dark Pounder a flying lesson.

The only thing that could drown out the cheers of the crowd, was the clash of steel on steel. The smell was unearthly, as were the sights. Bent steel, burnt motors, shattered plastic, flaming butane, and ripped aluminum; all were forged into one great machination created simply to overload the senses.

On February 4, 2006, 13 teams hauled their robots out to Mike's Hobby Shop to face off for the top prize at SWARC'S Most Extreme Robot Challenge.

A member of Team Dragon Robotics after winning the Rookie Driver Award.

As always, safety was top priority and the staff made sure that the Robot Fighting League rules were followed to the letter. By the time all the robots were deemed safe according to the RFL guidelines, SWARC'S Most Extreme Robot Challenge was off to a smooth start. The schedule was simple: fight bots until 1:00 P.M., and then take a lunch break for 30 minutes. After the break, come back and finish the fight brackets to determine the winners.

Though there was a slight shortage of larger robots, a round-robin configuration allowed for plenty of destruction. For the ants, a simple double elimination format was used. All robots weighing over a pound fought in the 24' by 24' Texas Toad Tank, while the ants dished out damage inside of the 4' by 4' ant box.

The ants (1-lb max.) kicked off the Most Extreme Robot Challenge with 10 bots and tons of hits rounding out the class. In addition to the usuals like Team Dark Forces' vertical spinner Dark Pounder were some newcomers like T.U.F.Ti.C.K. and Hungry Hungry Hippo. After proving their worth in battle, two drum spinners met in the finals. Fear Factor of Team Probotics and the aforementioned Team Wapu's Hungry Hungry Hippo. After a great fight Hungry Hungry Hippo came out on top as Texas' 1-lb champ.

Team Shiny Thing is ready for battle.

Four beetles (3-lb max.), two pushbots, and two spinners attended the event, but the group was hardly unexciting. The arrival of Dark Slayer — a titanium bladed spinner — assured plenty of carnage was to be had. In the finals, Dark Slayer came out on top by beating down Fear Factor Sr. — a vertical spinner with a 12-inch blade.

Weblinks: **www.Robotrebellion.net**
www.SWARC.org • www.Buildersdb.com

One spinner, one rammer, and one dustpan made up the hobbyweight (12-lb max.) class at SWARC'S Most Extreme Robot Challenge. Fulton's Folly and Scoopula pounded it out in the finals after defeating Mad Puppy — a horizontal spinner with a screaming gasoline engine.

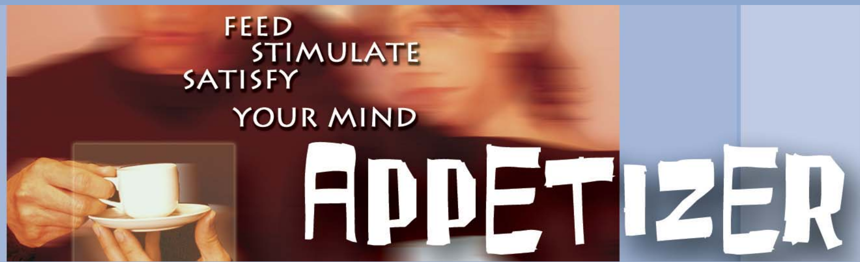
In the Featherweight (30-lb max.) class, Team Probotics' flame-spewing Lexan lifter My Girl Robot faced off against Team Monster's horizontal spinner Metal Fatigue. Two frightening fights later Metal Fatigue came out on top, proving to be the tougher of the two.

Crocbot, a past winner of many SWARC competitions, came out to protect his lightweight (60-lb max.) title against an all-steel wedge named Spinnerbait Jr. and the disk-wielding Guillotine. Three earth-shattering encounters left Crocbot on top for a third time, proving that, at the end of the day, it all comes down to how hard your shell is.

If you want to see some great fights, meet some great people, and have a ton of fun, come down to Texas for SWARC's next event — the Texas Cup on September 9th. You won't be disappointed!

Who said robot builders were crazy?

Formed in 2000, the Southwestern Alliance of Robotic Combat began as a simple group of robot enthusiasts. Since then, the gathering has increased and includes people from greatly differing backgrounds. SWARC holds their events in Mike's Hobby Shop where builders can buy spare parts. Mike's Hobby Shop is also an ideal home for Fuzzy's Toad Tank, which is now being used by SWARC.



The Robots Are Here! ... Well, Almost

by Jonathan E. Fant, P.E.

These are exciting times for robot enthusiasts. It is hard to avoid robots. This summer, COG from MIT is residing here in Portland, Oregon (pronounced "ORE-UH-GUN" for you East Coast folks) at the Oregon Museum of Science and Industry.

At least weekly there is an article in a newspaper or on TV regarding another robot (besides Honda's, that is). And, of course, the DARPA Grand Challenge was awesome this year.

The next DARPA Grand Challenge promises to be even more exciting next year as autonomous vehicles move to the urban street environment.

Locally, the IEEE Robotics group is sponsoring a software challenge in August to use video cameras in real time for indoor robot navigation functions.

On the education front, the FIRST and LEGO-based competitions are exposing more students to robotics all the time. Almost every university and local community college has some form of robotics involvement, whether academic instruction or a local student club.

The present military value of robots is said to be enormous, even with the limited capabilities of today's machines. The US government is spending large amounts of money on new robot designs, so further advances are expected in military robots.

Robots have become a national goal for many high-technology countries.

Obviously, all the pieces are coming together for robots now. When I think about the slow progress from simple manufacturing arms of the 60's to where we have come today, it is obvious that a lot of technical progress has been made.

These are definitely exciting times for robots!

But Hold on Now ...

It is easy to get carried away in all the technical details and advertising hoopla. Almost none of the journalists have a clue about what a robot is from a technical point of view (the fine folks at *SERVO* being an obvious exception). Often the latest robot is touted as being a "break-through" when it is not. Are all of these machines really robots?

Past authors of this column have discussed what a robot is and what is still lacking. Most readers of *SERVO* should already have a good idea of what a robot is, so I will give a condensed version in a moment.

First, here is what a robot is not. Often robot definitions I have seen are similar to this:

"A robot is a mechanical assemblage with moving parts and a controller, processor, or computer attached."

This is not an accurate definition since it includes just about everything made with moving parts, including your car and an elevator.

Another definition often heard is intended to limit the type of devices on a control basis:

"A robot is a mechanical assemblage with moving parts and a controller, processor, or computer attached that executes operations or movements with little or no human direction."

This one is getting closer, but fails what I call "The Elevator Test," since you can substitute "elevator" for "robot" in the definition and still make a true statement.

Something is Missing

So, something is still missing — something that we all intuitively know — that makes up the essence of a robot.

First, let's distinguish between robots and robotic devices.

Robotic devices include manufacturing arms, automatic car wash machines, elevators, the Roomba and Scooba products by iRobot, and even your late-model car (or the SUV you are trying to trade in for a hybrid).

So let's change the definition above to:

"A robotic device is a mechanical assemblage with moving parts and a controller, processor, or computer attached that executes operations or movements with little or no human direction."

As a definition for robotic devices, this one is pretty good, but does not define robots.

So here is a short definition of a robot:

"A robot is a robotic device that moves and operates in at least one existing real-world environment without tracks or guides and looks like a robot."

The last part is not optional, since modern commercial aircraft are robotic devices that operate without tracks or guides. The "existing real-world environment" is intended to separate out special-environment machines.

It is also circular and assumes one knows what a robot looks like. It does not state what part transforms the mechanical assemblage that looks like a robot into the true robot.

So, I leave it to another time or others to perfect the definition of a true robot. But it does give us insight into what is needed — machines that operate in existing environments with little or no human direction.

The Missing Parts

Sensors and software are the parts that transform a robot-looking mechanical assemblage into a robot. Sensors are required to give surrounding environment information to the software.

Without proper software, the hardware will look like a robot but remain a remote-controlled or pre-programmed puppet.

Honda's robot comes to mind here. It definitely looks like a robot, but requires a human to program specific step sequences or operate it by remote control (my opinion based only on observation since I cannot afford to buy one).

If one assumes that existing sensors are adequate, then software becomes the only missing piece. More specifically, some form of machine-intelligence robot software.

Certainly, all modern robot designers include software as a critical design component, but they typically focus in-depth on hardware, and then write just enough software to operate the hardware. This makes a great robotic device, but fails to make a robot.

Robot software must be real-time aware of the local environment, including interacting with humans, and be goal-oriented to provide useful functions for humans. Obviously, the robotic device component (mechanics of the robot) must work under the direction of the robot software.

Only when we focus on the machine-intelligence portion of our software will we finally build robots. We do not need to build C3P0 software to get started, just enough to build safe, useful robots.

This effort has already started. Stanford's DARPA team used a software-focused approach to win, the IEEE Video Odometry challenge (August 2006) focuses on a significant piece of the missing robot software, and the LEAF project (and others) are already underway.

This is the age of robots ... now let's get busy and write robot software! **SV**

AUTHOR BIO

*Jonathan E. Fant, P.E. is president and founder of Future Robotics, Inc. (**www.futurerobotics.com**) — a company dedicated to building real-world robot hardware, software, and accessories. He is an electrical engineer with a background in electrical power, controls, and robotics. He can be reached at jon@futurerobotics.com*

Then and NOW

ROBOTS WHO CARE FOR PEOPLE

by Tom Carroll

Again, I use the word *Who* in the title of my article, just as in my previous article about *Robots Who Live With People* in the April '06 issue of *SERVO*. I like the word, as it implies a closer relationship to a type of machine we all appreciate so much. However, there is a wide gap between *living with* and *caring for* a person — a difference that has been apparent for many millennia.

Centuries ago, caring for a person who was disabled or ill usually meant keeping them comfortable in a bed and feeding them until they healed or died. Sometimes the "healing" process was worse than dying a peaceful death. Stretchers were used to move a sick person from one point to another. Later, as the science of medicine progressed and hospitals were developed to care for people, caring became more refined.

Hospitals began to develop mechanical means to move or position a sick person instead of "man-handling" them about. Wheelchairs were the first of these devices, as the transport of an ill patient was very important. Later, manual lifts were developed to move a person from a bed to a wheel chair or rolling gurney, but human muscle power was still required.

As the twentieth century progressed, medical technology advances created powered patient lifts that rolled along on overhead tracks. A sling was placed around the patient and an overhead electric winch and cable attached to the sling allowed a nurse to lift the person out of a chair, bed, toilet, or bath. The winch was attached to the overhead track and a person could move the patient anywhere the tracks were mounted by just pushing the person along with their hands. Figure 1 shows such a patient lift in use.

A person is usually still required to operate the lift and move the patient along the overhead track — much like a butcher moving a side of beef in the cutting room. Powered wheelchairs and scooters allow people with disabilities to move about on their own — a joystick is at the end of one armrest controlling the direction and speed of the chair. However, when the disability is great enough, a person is sometimes required to move the patient into and out of the wheelchair and into a bed, bath, or toilet. At this point, the person loses their ability to live independently. Figure 2 shows an electric lift that is attached to the side of stairs to assist people in climbing stairs.

A Robot to Assist People in Daily Life

Living independence is a situation that I have examined thoroughly. Several years ago, my son Tom was severely injured when he was in the Navy and required constant attention for all of his basic needs. He had been a physically fit Navy Seal candidate, but was reduced to helplessness after an accident. Fortunately, this condition passed after six months or so, but I had been thinking about what type of robot I could build that would attend to his needs.

At about the same time, my sister Liz was slowly becoming an invalid due to a neurological disease and would require an entirely different type of care — robotic or otherwise. Just what kind of robot could I develop that would satisfy *both* of their needs? There are so many different types and degrees of disabilities that no one design can satisfy all needs. I was having a hard time coming up with a functional design for a robot to assist every type of disability.

Shortly after this initial design, I got to talking with a delightful 98-year-old lady who knew of my robotics background. Myra was living independently, but had to have a housekeeper come to her home daily to help her with a few basic needs. Her mind was sharp, as she kept track of her stock portfolio and was a great conversationalist, but found she needed basic help in moving and doing things around her home.

After a particularly difficult morning, I happened to go by her home for a visit. "Tom," she asked me, "I wish you could build a robot that could help me live day-to-day without having to

Figure 1. Overhead lift.

Figure 2. Stairlift.

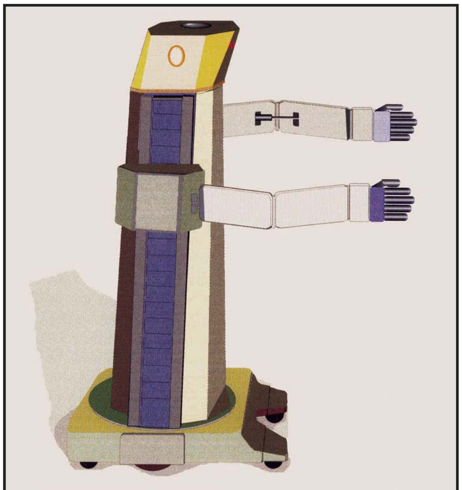


Figure 3. One of my robot drawings.

have a person coming by all the time to help me do the simplest things that I used to do for myself. I don't want a person hanging around me and going through all my things."

A Personal Assistant Robot for the Elderly

Myra passed away just short of 100 years old, but I thought about her desire for a long time. I dug out the data I compiled a dozen years ago when I was part of a panel that was tasked to determine just what people wanted in a home robot. Much of the information also applied to a robot to care for the elderly, but the basic *form* of the robot turned out to be entirely different. I soon discovered that a robot to assist the elderly could be a single design that would pretty much cover many of the needs of all seniors.

The elderly generally have all their motor functions. These functions just don't work as well as they did in the past. They can reach out to grasp something, but it may be painful and with limited grasping ability. They can get out of and into a chair, but that task may be difficult. We've all joked about the advertisement that shows an elderly person calling out, "Help, I've fallen and can't get up," into a *Life Alert* pendant, but many elderly people have lain upon a floor for hours until help arrived. I have used the term *Personal Assistant Robot* for years to describe a robot that can actually *physically assist* a person, not just a "go-and-fetch" machine or a mechanized conversationalist.

In 1995, I visited an upper-scale nursing care facility in Long Beach, CA, and spoke with 47 patients with all levels of assistive needs. As I expected, there were some (14%) who were pretty much against having a "mechanical servant" in their home, even if they could return to their homes. I can understand this reaction, seeing as how it is so new and unknown. It was the completely bed-ridden patients who expressed this sentiment more than the more mobile and active residents did.

However, over 68% of those with whom I spoke were very enthusiastic about having a robot to assist them in daily needs. I showed all of them — 12 men and 35 women — two drawings of the robot I had designed and described how it worked and how it could help them (see Figure 3). I spoke honestly, telling them that the robot was not a house-cleaning maid, but a constant companion at their beck and call. The remaining 18% took a "wait-and-see" stance. I was told by almost all of the patients to, "Hurry up and build this thing. We [all seniors] need it now."

They explicitly stated that they did not want a robot that followed them around like a lost dog, asking them if they'd taken their pills or similar reminders. Cordless phones and radio pendants for around their necks can summon their doctor, the police, or other emergency services, and timed pill containers with a voice reminder can easily keep seniors on their medications at a far cheaper cost. What they all desired was a robot *who* could *physically do something* — not just roll around and "beep and talk."

They wanted a robot servant *who* could assist them from the floor after a fall. They wanted a robot *who* could be directed to go into the kitchen, take a Lean Cuisine beef and broccoli meal out of the freezer, place it in the microwave oven for six minutes, and bring it to them when it was done. They wanted a robot *who* could help them into and out of a chair, bed, or even a toilet, safely.

All of these tasks are pretty hard for a robot of today, especially the *safely* part. In researching the *FDA Code of Federal Regulations* under part 890, "Physical Medicine Devices" it appears that sub part F, "Physical Medicine Therapeutic Devices," part 890.5050, describes a "Daily Activity Assist Device" under which personal assistant robots might have to conform for safety reasons.

Not surprisingly, many of the seniors whom I spoke with wanted the robot to clean the house, though I had mentioned that the robot I had designed was *not* a maid. They had no idea just how difficult house cleaning is – *thorough* house cleaning. The I-Robot *Roomba* and *Scooba* do a fairly good job of daily carpet and floor cleaning, but dirt and dust does not just accumulate on the floors. It is deposited upon furniture, walls, and all over kitchens and bathrooms. No doubt that some of these seniors have since purchased a Roomba.

A competent maid can spend an hour a week doing a far better job of cleaning a home than any robot, and he or she can also bring in a week's load of groceries and place them in cabinets, refrigerators, and freezers, oriented so that a robot can read the barcodes and retrieve them (with a bar-code or RFID reader in the robot's claw). I shudder at the thought of a robot set loose in an elderly person's home trying to dust a shelf covered with many crystal and ceramic figurines. At maybe \$50-\$100 a week for simple house cleaning and food arrangement — depending on the home's size — a senior can pretty much live independently with a well-designed and functional personal assistant robot.

Let a person do these low-paying tasks for an hour or two a week and let the robot assist a person the other 166 hours. This is certainly less expensive than a daily, full-time caregiver in the home at a cost of \$150 to \$200 a day, or \$55,000 to \$73,000 a year. A nursing home can be significantly higher.

Personal Assistant Robot Configuration

Not surprising for people who know me, my design uses two SCARA arms that ride up and down on two tracks on each side of the robot. They are leadscrew-driven (actually — recirculating ballscrews for efficiency) and can

lift close to 200 pounds each, much like we would move our arms about on the top of a chest that is as tall as our upper chest.

Figure 4. Engelberger beside his HelpMate.

Figure 5. CAD model.

Figure 6. Care-O-Bot.

Figure 7. Care-O-Bot Manipulator.

SCARA configuration robot arms have all vertical axes and operate in a horizontal plane and take little power to move a payload back and forth, just as a fingertip can move a heavy door back and forth. SCARA configurations offer great lifting power, but have a problem reaching up into or down in cavities to retrieve objects without the addition of special end effectors at the end of the arms. Most industrial SCARA robot arms have a special end-effector that moves up and down for precision parts insertion and similar applications.

Most designers of personal robots to assist the elderly have used the typical human arm configuration — a modified version of the industrial robot's "revolute configuration." The shoulder joint is at the top of the robot's chest structure with a lower elbow connected to a forearm and "hand." This type of arm can easily reach down into a cavity and still reach up over the robot's head, but requires a greater amount of power to accomplish a lifting task. I wrote more extensively about personal robot design in the April '06 issue of *SERVO*.

Another design issue was the robot's ability to keep from tipping over while handling a human being. Extendible "feet" seemed to partially solve this problem. Many other hurdles must be met, such as reliable speech understanding, sophisticated software to make sense of a series of verbal commands, useful vision systems, and many other design problems.

How much would such a robot cost? As with anything new, the costs will probably be far higher that any designer ever imagined. Just as the first cell phones were as big as a brick and cost several thousand dollars, they are now almost invisible and are "given away" with your wireless phone service. The first personal assistant robots will probably be heavier than we'd like and cost less than a Mercedes, but more than a Chevy. Using as many "off-the-shelf" items as I could, I soon realized that basic metal forming, machining, circuit board populating and soldering, plus assembly of complex mechanisms quickly brought the cost of initial robots skyward.

As with any complex design project, prototypes must be built, trials run, issues addressed, and government regulations covered. The final product must be engineered, manufactured, marketed, and delivered. When I first moved up to the Seattle, WA area, I met a man who was quite interested in providing the large amount of venture capital necessary to get the personal assistant robot project underway. After the "dot.com crash" and its devastating effect on new tech start-ups, he had to drop out of the funding after another firm ate his funds. I have temporarily shelved the project, though I am constantly asked about its status.

Joe Engelberger Changes His Focus on Robotics

Stepping aside from my own design ideas, I'd like to talk about other personal robots that are making their way into our homes. Joe Engelberger — world-renown as the "Father of Industrial Robotics" — has changed his focus in the last decade from factory robots to robots who care for the elderly. Shown next to his successful HelpMate robot designed for hospitals (Figure 4), Joe is pushing close to 81 years old himself.

I had the privilege to eat a quick lunch with Joe back in 1984 at the International Personal Robot Congress in Albuquerque, NM — one of the first *personal* robot symposiums. Joe was already thinking ahead of a robot for the home, long before his HelpMate hit the hospital floors.

He clearly sees the need for a robot in the homes of our seniors. In several interviews with TV and magazine people, he has said that his "HomeMate will fetch and carry, cook and clean, and help with all the activities of daily living for a fraction of the cost of skyrocketing nursing home or in-home care; leasing at $1 an hour, or $600 a month."

Figure 5 is a CAD drawing of his proposed HomeMate robot. "The robot will have thick skin," he chuckled. "With an 85-year-old person, the robot will have to react to four or five questions: 'What is lunch?' 'Where is it?' It's very easy to sense distress. It can go up and say, 'Tell me about your high school reunion party,' and the robot is going to clap every time." Engelberger had proposed the creation of a company called RoboCare, Inc. (not to be confused by the Italian consortium of the same name), to design and produce his robot. Quite frankly, if anyone can do it, Joe can.

To be expected, US companies are not the only firms interested in

Figure 8. RI-MAN holding a full-sized human doll.

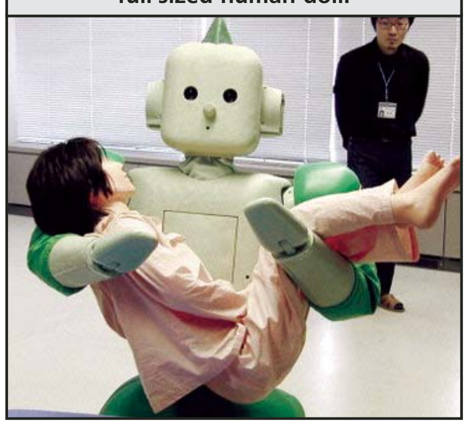


Figure 10. Sogang University robot.

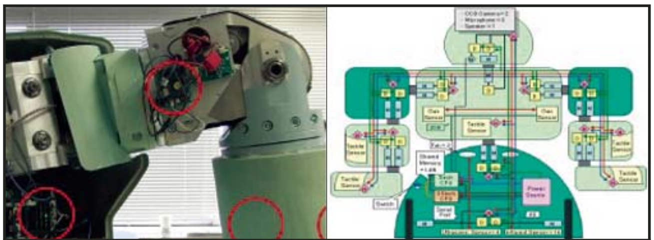


Figure 9. RI-MAN schematic.

producing a robot to assist the elderly in independence. Fraunhofer IPA, of Stuttgart, Germany has produced two demonstrator platforms of a robotic home assistant called Care-O-Bot. The first model — Care-O-Bot I — was strictly a mobile platform for research purposes. It had a touch LCD screen for basic input/output information. The Care-O-Bot II model is equipped with adjustable walking supporters to assist a person when walking behind the robot. The robot also has a fairly sophisticated manipulator arm attached.

Figure 6 shows the Care-O-Bot II with its amazing manipulator arm. It can navigate autonomously in indoor environments and the manipulator can be programmed for various pick-and-place tasks. The control software of the Care-O-Bot II operates on two industrial PCs and a separate hand-held control can be used for remote programming and operations.

Autonomous functions can operate the manipulator arm, allow roving about a home, utilize images from a camera and laser scanner in the head, and be controlled by voice commands or computer keyboard inputs. Figure 7 shows the manipulator in use.

At an automation fair in Germany in 2002, the Care-O-Bot II actually was programmed to approach visitors and hand out business cards. The robot then asked the visitor for his or her business card and placed them in a tray.

It goes without saying that the greatest interest in robots and the home of the largest robot manufacturers is in Asia — in particular Japan and Korea. From Sony's Aibo to Honda's Asimo, the latest advances in personal robotics has come from Japan. Though Sony has recently bowed out of the field, many new products are arising from these countries. It is estimated by the Japan Robot Association that over 16,000 "service robots" — the term for non-industrial, but also not entertainment robots — are in place, with the majority in hospitals and nursing homes.

Japan and Korea are facing a greater percentage of elders and a subsequent greater need for care. The RIKEN-Bio-Mimetic Control Research Center in Nagoya, Japan has developed RI-Man — an anthropomorphic (man formed) robot to be used as an elder care robot. Figure 8 shows the RI-MAN carrying a human-sized doll, weighing only 26 pounds though. At 5' 2" tall, the 200-pound robot is expected to carry 70 KG — over 150 pounds — in the near future. Covered with a pliable, soft foam rubber skin, it is expected to be people-friendly. Figure 9 shows two of the robot's joints and an overall schematic.

Researchers at Sogang University recently developed a robot to assist disabled and elderly people with basic mobility. Figure 10 shows how a person is attached to the robot's arms at their waist, and the machine slowly leads the person about a room. Japan's Waseda University has long led the world in specialized robot development, but MIT in the US with its COG and KISMET personal robots is not to be left out of the game. Carnegie Mellon and other US universities always manage to keep us amazed with some very unique robots.

I have only touched upon a few of the new robots designed to care for people. There are many US companies and private researchers developing new machines. Interest in robots for the home is experiencing a tremendous rise. The movie *Bicentennial Man* and similar films have shown robots in the homes of average citizens. Non-tech people are scooping up Roombas and Scoobas.

There are always the "early adapters" who will buy some seemingly useful gadget just to be the "first person on the block" with something new. Most people are the "middle and late adapters" who want a new technology to prove itself before they buy. Early adapters "pay the price" and help the companies weed out the kinks and problems before the masses move in to buy. Home care robots for the elderly will soon be at the marketplace. **SV**

Advertiser Index

All Electronics Corp. 25, 35
Critical Velocity 35
CrustCrawler 9
Dimension Engineering 22
DynoMotion 35
Futurlec 35
Future Robotics 63
Garage Technologies 35
Hitec 3
Hobby Engineering 66
Industrial Ventures 66
Jameco Robot Store 2
Lemos International Co., Inc. 13
Lorax Works 25
Lynxmotion, Inc. 23
Madell Technology Corporation 35
Maxbotix 35
Medonis Engineering 35
MeerKat Systems, Inc. 35
Net Media 83
NU-BOTICS 25
Parallax, Inc. ....Back Cover
PCB Fab Express 63
Pololu Robotics & Electronics 56
Robot Power 56
Robot Shop 78
Solarbotics....17
Solutions Cubed 34
SpectroTech 35
Technological Arts 35
Vantec 56